Rethinking the Science of Statistical Privacy

Changchang Liu

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Electrical Engineering
Adviser: Professor Prateek Mittal

April 2019
Abstract

Nowadays, more and more data, such as social network data, mobility data, business data, medical data, are shared or made public to enable real world applications. Such data is likely to contain sensitive information and thus needs to be obfuscated prior to release, to protect privacy. However, existing statistical data privacy mechanisms in the security community have several weaknesses: 1) they are limited in protecting sensitive information in the static scenario, and can not be generally applied to accommodate temporal dynamics. With the increasing development of data science, a large amount of sensitive data such as personal social relationships are becoming public, making the privacy concerns of a time series of data more and more challenging; 2) these privacy mechanisms do not explicitly capture correlations, leaving open the possibility of inference attacks. In many real world scenarios, the data tuple dependence/correlation occurs naturally in datasets due to social, behavioral and genetic interactions between users; 3) there are very few practical guidelines on how to apply existing statistical privacy notions in practice, and a key challenge is how to set an appropriate value for the privacy parameters.

In this thesis, we aim to overcome these weaknesses to provide privacy guarantees for protecting dynamic data structures, dependent (correlated) data structures. We also aim to discover useful and interpretable guidelines for selecting proper values of parameters in the state-of-the-art privacy-preserving frameworks. Furthermore, we investigate how an auxiliary information – in the form of prior distribution of the database and correlation across records and time – can influence the proper choice of the privacy parameters. Specifically, we 1) first propose the design of a privacy-preserving system called LinkMirage, that mediates access to dynamic social relationships in social networks, while effectively supporting social graph-based data analytics; 2) explicitly incorporate structural properties of data into current differ-
ential privacy metrics and mechanisms, to enable privacy-preserving data analytics for dependent/correlated data; and 3) finally provide a quantitative analysis of how hypothesis testing can guide the choice of the privacy parameters in an interpretable manner for differential privacy and other statistical privacy frameworks.

Overall, our work aims to place the field of statistical data privacy on a firm analytic foundation that is coupled with the design of practical systems.
Acknowledgements

Firstly, I would like to express my sincere gratitude to my advisor Professor Prateek Mittal for his tremendous academic support that makes my PhD an amazing experience. In particular, I am very grateful to him for enlightening me, and introducing me to my first glance of research in security and privacy. He encouraged me to focus on research topics that I am most interested in, while at the same time providing me useful insights and suggestions that motivated me to widen my research from various perspectives. I have always enjoyed our weekly meetings where his advice on both research as well as on my career have been priceless in helping me to grow as a security/privacy research scientist. His immense professionalism, motivation, and patience helped me in all of my research and the writing of this thesis. Other than his valuable academic support, I also wholeheartedly appreciate all his help when I had crisis in my general life. I could not have imagined having a better advisor and mentor for my PhD study.

Besides my advisor, I would like to thank the rest of my thesis committee: Professor Niraj Jha and Professor Peter Ramadge for their support and encouragement. I also want to thank the rest of my thesis readers: Professor Sun-Yuan Kung and Dr. Supriyo Chakraborty for their brilliant comments and suggestions. I would also like to appreciate the help from Professor Mung Chiang and Professor Paul Cuff for serving as committee members in my general exam a few years ago. At that time some of my works were still in a very early stage, the feedback they provided has motivated the developments of several important parts of this thesis.

My sincere thanks also goes to Dr. Supriyo Chakraborty and Dr. Seraphin Calo, who provided me an opportunity to join their team as an intern. They have been
truly dedicated mentors. I really appreciate their support when so generously hosting me in IBM.

I also want to thank my fellow labmates in Princeton INSPIRE Lab for the stimulating discussions, for the sleepless nights we were working together before deadlines, and for all the fun we have had in the last five years. They are Peng Gao, Yushan Liu, Yixin Sun, Sameer Wagh, Thee Chanyaswad, Arjun Bhagoji, Hooman Moghaddam, Liwei Song, Arsalan Mosenia, and Daniel Cullina. I would also thank Professor Shouling Ji in Zhejiang University, Professor Xi He in the University of Waterloo and Dr. Shiqiang Wang in IBM Research for their help in my research. I also want to thank Dr. Fangfei Liu who shares with me a lot of great experience in research and life. I would also thank Zhixing Xu for all the support he has shown to my family. I thank all my friends in the United States and China for their companionship and help.

This dissertation would not have been possible without the funding support from the Charlotte Elizabeth Procter Fellowship, the IBM PhD Fellowship, School of Engineering Award for Excellence, Electrical Engineering PhD Early Career Award, the Anthony Ephremides Fellowship, First-year PhD Student Fellowship, and the National Science Foundation (NSF) under the grant CNS-1409415, CNS-1423139, CNS-1553437 and CCF-1617286, an Army Research Office YIP Award, and faculty research awards from Google, Cisco, Intel, and IBM.

Much thanks to the staff in the Princeton Electrical Engineering Department, Colleen Conrad, Roelie Abdi, Lori Bailey, Stacey Weber, Sania Sadhvani and Lisa Lewis, for their enormous help.

I would especially like to thank physicians and nurses at Princeton University McCosh Health Center and the Penn Medicine Princeton Medical Center. They have
been there to support me when I was diagnosed with a cyst in 2016 and also when I had a hard time in my pregnancy and delivery process in 2017. Thanks a lot for all their help for keeping me and my son Max healthy.

Last but not the least, I would like to thank my family: my father Xunliang, my mother Lanling and my sister Jieyuan for supporting me spiritually throughout writing this thesis and my life in general. My parents always gave me the freedom to make my own choices and allowed me to become the person I wanted to be. Most importantly, they are always there for me, especially during the past two years which have been the toughest time in my life. They came to the United States several times to take care of me during my recovery from the cyst surgery and from the baby delivery. I really appreciate their support and sacrifice in helping me overcome these hard times. I would also want to thank my husband Wei-Han for his unconditional love and for giving me a happy family. Especially for the recent two years, I have witnessed the perseverance and resilience in him which gives me the most courage and strength to overcome obstacles. Finally, I would thank my son Max who is always my strongest support and my greatest happiness, and I dedicate this thesis to him.
Contents

Abstract .................................................................................. iii

Acknowledgements ................................................................. v

List of Tables ........................................................................... xvi

List of Figures ........................................................................... xvii

1 Introduction ........................................................................... 1

1.1 Motivation ........................................................................... 1

1.2 Overview of the Dissertation ...................................................... 6

1.3 Dissertation Contributions ...................................................... 7

1.4 Dissertation Organization ...................................................... 9

2 Related Work ......................................................................... 11

2.1 Privacy-preserving Graph Data Publishing ................................. 12

2.1.1 Graph Privacy with Labeled Vertices ............................... 13
2.1.2 Graph Privacy with Unlabeled Vertices ................................ 14
2.1.3 Preserving Privacy for Social Networks ................................. 15

2.2 Differential Privacy Frameworks ........................................... 17
  2.2.1 Achieving Differential Privacy ........................................ 18
  2.2.2 Categories of Differential Privacy .................................... 19
  2.2.3 Implementation and Analysis of Differential Privacy ............... 20
  2.2.4 Differential Privacy under Adversaries' Auxiliary Information .... 21

2.3 Hypothesis Testing and Differential Privacy ............................... 23
  2.3.1 Hypothesis Testing ................................................... 23
  2.3.2 Hypothesis Testing in Differential Privacy .......................... 25
  2.3.3 Setting the Privacy Budget ......................................... 26

3 LinkMirage: Enabling Privacy-preserving Analytics on Correlated
and Evolving Graph Data ....................................................... 28
  3.1 Motivation ................................................................. 29
  3.2 System Architecture and Threat Model .................................. 33
  3.3 System Overview and Roadmap ......................................... 34
    3.3.1 LinkMirage System ............................................... 34
    3.3.2 Privacy Evaluation ............................................... 35
3.3.3 Utility Evaluation

3.4 LinkMirage System
    3.4.1 Design Goals
    3.4.2 LinkMirage: Deployment
    3.4.3 LinkMirage: Perturbation Algorithm
        3.4.3.1 Dynamic Clustering
        3.4.3.2 Selective Perturbation
    3.4.4 Scalable Implementation
    3.4.5 Visual Depiction
    3.4.6 Supporting Applications

3.5 Privacy Analysis
    3.5.1 Experimental Datasets
    3.5.2 Anti-Inference Privacy
    3.5.3 Indistinguishability
    3.5.4 Anti-aggregation Privacy
    3.5.5 Relationship with Differential Privacy
    3.5.6 Summary for Privacy Analysis

3.6 Applications
3.6.1 Anonymous Communication ........................................ 62
3.6.2 Vertex Anonymity ......................................................... 63
3.6.3 Sybil Defenses ............................................................. 65
3.6.4 Privacy-preserving Graph Analytics ................................. 66
3.6.5 Summary for Applications of LinkMirage .......................... 66
3.7 Utility Analysis .............................................................. 67
3.7.1 Metrics ............................................................... 67
3.7.2 Relationships with Other Graph Structural Properties .......... 69
3.7.3 Upper Bound of Utility Distance .................................... 70
3.7.4 Utility Experiments Analysis ......................................... 72
3.8 Chapter Summary ........................................................... 72
4 Dependence Makes You Vulnerable: Differential Privacy Under Dependent Tuples ........................................ 73
4.1 Motivation ................................................................. 74
4.2 Adversarial Model .......................................................... 78
4.3 Inference Attack: Differential Privacy under Dependent Tuples ........................................ 80
4.3.1 Dataset Description ....................................................... 81
4.3.2 Differentially Private Data Release ................................ 83
4.6.2.1 Clustering ........................................ 109
4.6.2.2 Classification ................................. 110
4.6.2.3 Degree Distribution ............................. 111
4.6.3 Summary for the Experimental Analysis .......... 112
4.7 Chapter Summary ........................................ 113

5 Investigating Statistical Privacy Frameworks from the Perspective of Hypothesis Testing ................................. 114
5.1 Motivation ........................................ 115
5.2 Methodology Overview ............................... 120
5.3 Quantification of DP from the Adversary’s Hypothesis Testing .................. 125
  5.3.1 Quantification of Unbounded DP .................. 125
    5.3.1.1 Hypothesis Testing Problem ............... 125
    5.3.1.2 Decision Rule .............................. 126
    5.3.1.3 Evaluating Hypothesis Testing Performance .... 129
    5.3.1.4 Choosing Proper $\epsilon$ ..................... 131
    5.3.1.5 Plausible Deniability Property ............. 133
  5.3.2 Quantification of Bounded DP .................... 135
  5.3.3 Quantification of ($\epsilon, \delta$)-DP .................. 136
C Deriving $F_{\beta_{score}}^*$ under Different Scenarios and Proving Theorem 22

C.1 Deriving $F_{\beta_{score}}^*$ under Unbounded DP .......................... 163
C.2 Deriving $F_{\beta_{score}}^*$ under Prior Distribution ......................... 165
C.3 Deriving $F_{\beta_{score}}^*$ under Record Correlation ......................... 167
C.4 Deriving $F_{\beta_{score}}^*$ under Temporal Correlation ....................... 167
C.5 Proof for Theorem 22 ................................................................. 168

Bibliography ............................................................................... 170
List of Tables

3.1 Notations Used in Chapter 3 .......................................................... 32
3.2 Temporal Statistics of the Facebook Dataset ................................... 45
3.3 Temporal Statistics of the Google+ Dataset ................................... 48
3.4 Modularity of Perturbed Graph Topologies ................................... 65
3.5 Graph Metrics of the Original and the Perturbed Graphs for the Google+ Dataset .......................................................... 69
4.1 Notations Used in Chapter 4 .......................................................... 77
5.1 Notations Used in Chapter 5 .......................................................... 119
5.2 The Maximal $\epsilon$ under a Required Bound of $F_{\text{score}}$ ........... 133
List of Figures

3.1 **LinkMirage architecture.** LinkMirage first collects social link information through our social link app or directly through the OSN providers, and then applies an obfuscation algorithm to perturb the original social graph(s). The obfuscated graph(s) would be utilized to answer the query of the untrusted applications in a privacy-preserving manner. The third-party application (which queries the social link information) is considered an adversary which aims to obtain sensitive link information from the perturbed query results. 33

3.2 **Our perturbation mechanism for** $G_t$. Assume that $G_{t-1}$ has already been dynamically obfuscated, based on dynamic clustering (step 1) and selective perturbation (step 2). Our mechanism analyzes the evolved graph $G_t$ (step 3) and dynamically clusters $G_t$ (step 4) based on the freed $m$ hop neighborhood ($m = 2$) of new links (between green and blue nodes), the merging virtual node (the large red node in step 4), and the new nodes. By comparing the communities in $G_{t-1}$ and $G_t$, we can implement selective perturbation (step 5), i.e. perturb the changed blue community independently and perturb the unchanged red and green communities in the same way as $G'_{t-1}$, and then perturb the inter-cluster links. 39
3.3 **Visualization of LinkMirage on Facebook Dataset** for $t = 3, 4, 5$.

On the left, we can see that LinkMirage has superior utility than the baseline approach (Mittal et al.), especially for larger values of $k$ (due to dynamic clustering). On the right, we show the overlapped edges (black) and the changed edges (yellow) between consecutive graphs: $t=(3, 4)$ and $t=(4, 5)$. We can see that in LinkMirage, the perturbation of unchanged communities is correlated across time (selective perturbation), minimizing information leakage and enhancing privacy.

3.4 (a), (b) represent the link probability distributions for the whole Facebook interaction dataset and the sampled Facebook interaction dataset with 80% overlap. We can see that the posterior probability of LinkMirage is more similar to the prior probability than the baseline approach.

3.5 **Link probability distribution for the Google+ dataset under the adversary’s prior information extracted from the social-attribute network model in [50].**

3.6 (a), (b) represent the temporal indistinguishability for the whole Facebook interaction dataset and the sampled Facebook interaction dataset with 80% overlap. Over time, the adversary has more information, resulting in decreased indistinguishability. We can also see that LinkMirage has higher indistinguishability than the static method and the Hay’s method in [56], although it still suffers from some information leakage. Comparing (a) and (b), we can see that the advantage of LinkMirage is more obvious for larger overlapped graph sequence.
3.7 (a)(b) show the temporal anti-aggregation privacy for the Google+ dataset and the Facebook dataset, respectively. The anti-aggregation privacy decreases as time evolves because more information is leaked with more perturbed graphs available. Leveraging selective perturbation, LinkMirage achieves much better anti-aggregation privacy than the static baseline method. 58

3.8 The worst case probability of deanonymizing users’ communications \((f = 0.1)\). Over time, LinkMirage provides better anonymity compared to the static approaches. 61

3.9 (a) shows the false positive rate for Sybil defenses. We can see that the perturbed graphs have lower false positive rate than the original graph. Random walk length is proportional to the number of Sybil identities that can be inserted in the system. (b) shows that the final attack edges are roughly the same for the perturbed graphs and the original graphs. 64

3.10 (a), (b) show the utility distances using the Google+ dataset and the Facebook dataset, respectively. Larger perturbation parameter \(k\) results in larger utility distance. Larger application parameter \(l\) decreases the distance, which shows the effectiveness of LinkMirage in preserving global community structures. 71

4.1 The participation of one individual \(D_j\) would harm the privacy of another individual \(D_i\) based on the dependence between them. 78
4.2 Gowalla’s social dataset colored according to the results from K-means clustering of the location dataset. We can see that the location dataset is inherently correlated and the social dataset well represents such relationships. .......................... 84

4.3 The distance between the location vectors of users. ........... 84

4.4 Location vector inference attack. (a) represents Attack 1 un- der independent tuple assumption and (b) represents Attack 2 under dependent tuple assumption. ..................... 86

4.5 The inference error of the adversary’s inference attack. ..... 88

4.6 Separation for different queries under dependent tuples. . . 94

4.7 Comparison of (a) utility and (b) privacy performance of dif- ferent perturbation mechanisms. .............................. 108

4.8 Clustering accuracy under different perturbation methods. . 109

4.9 Classification accuracy under different perturbation methods. 110

4.10 Degree distribution accuracy under different perturbation methods. ......................................................... 111

5.1 Overview of our quantitative procedure to investigate the choice of an appropriate value of \(\epsilon\) from the perspective of the adversary’s hypoth- esis testing. ....................................................... 121

5.2 Adversary’s hypothesis testing of detecting a particular record under unbounded DP. By setting a threshold \(\theta\), the adversary’s decision rule is \(o_\theta\). ............................................................. 129
5.3 (a) PR-relation and (b) the highest $F_{\beta}$score of detecting a particular record under unbounded DP .................................................. 131

5.4 PR-relation of detecting a particular record from ($\epsilon, \delta$)-DP results with varying (a) $\epsilon$ and (b) $\delta$, respectively ........................................... 137

5.5 (a) Hypothesis testing and (b) PR-relation of detecting a particular record for adversaries with access to the prior distribution of the input data ($\rho_p = 0.2$) ......................................................... 141

5.6 (a) Hypothesis testing and (b) PR-relation of detecting a particular record for adversaries with access to record correlation ($\rho_p = 0.2, \rho_c = 0.1$) ................................................................. 145

5.7 (a) Hypothesis testing and (b) $F_{\beta}$score of detecting a particular record for adversaries with access to temporal correlation ($\rho_p = 0.2, \rho_c = 0.1, \rho_t = 0.1$) ......................................................... 146

5.8 The highest $F_{\beta}$score achieved by the adversary under different auxiliary information (setting $\beta = 1$, i.e., $F_{1}$score) ......................................................... 148

5.9 Comparison of neighboring databases in the state-of-the-art statistical privacy frameworks ................................. 150

xxi
Chapter 1

Introduction

This dissertation focuses on new challenges of statistical privacy and how to design novel privacy frameworks and mechanisms to tackle these challenges. In this chapter, we first introduce the motivation for this dissertation. Research on statistical privacy is mainly driven by the increasing development of big data in real world applications aiming at balancing privacy of users’ sensitive information and utility of enabling data-driven analytics. We then give an overview of the privacy frameworks and mechanisms proposed in this dissertation. We end the chapter by summarizing the contributions and outlining the organization of the dissertation.

1.1 Motivation

An increasing amount of data (such as social network data, mobility data in IoT systems, business data, medical data, etc.) is generated by computer networks and services. The sensitive nature of this data gives rise to significant security and privacy concerns. Data privacy is an issue of critical importance, motivating perturbation of
query results over sensitive datasets for protecting users’ privacy.

However, existing statistical data privacy mechanisms have several serious weaknesses where they 1) do not explicitly account for temporal dynamics of data, such as evolving network graphs, 2) do not explicitly account for the data tuple dependence/correlation that occurs naturally in real-world datasets and 3) lack practical guidelines on how to set appropriate values of the privacy parameters. In this thesis, we aim to overcome these weaknesses by proposing novel privacy mechanisms for protecting dynamic data structures and dependent (correlated) data structures, and providing useful and interpretable guidelines for selecting the privacy parameters in practice. Furthermore, we aim to analyze how the dynamic data structures and dependent/correlated data structures would affect the choice of privacy parameters.

**How to protect privacy under temporal dynamics of graph data?**

In this dissertation, we consider social networks as a representative example of graph data. Social network based trust relationships are often leveraged to improve system functionality and security. A key issue in the design of such systems, is the revelation of users’ trusted social contacts to an adversary – information that is considered sensitive in today’s society. Recent work has examined the privacy implications of such designs, and proposed approaches to protect the privacy of users’ social contacts in a static network. In this context, prior work has explored approaches for preserving link privacy in static social networks. These methods aim to randomize the structure of the social graph by introducing noise (by adding and deleting links), and making the perturbed graph available to applications. However, social network topologies are dynamic, and evolve over time with the addition and deletion of users and links – this introduces a new privacy chal-
lenge, as adversaries can combine information available in multiple perturbed graphs to infer users’ social contacts.

What are the implications of social network dynamics on (a) the privacy of users’ social contacts? and (b) the design of effective privacy-preserving mechanisms? How can we enable the design of social network based systems while preserving the privacy of users’ social contacts in a dynamic setting? Towards this end, we propose the design of LinkMirage system in Chapter 3 that mediates access to dynamic social relationships while enabling general graph applications. We focus on these challenges and aim to minimize the privacy degradation due to release of multiple network snapshots by introducing correlation in the perturbation process over time.

**How to protect privacy under general data dependence/correlation?**

Among the existing statistical privacy metrics in the literature, differential privacy (DP) has become the gold standard for a rigorous privacy guarantee \[34, 36, 39, 43\]. Simply stated, it guarantees that the distribution of query results changes only slightly due to the modification of any one tuple in the database.

Dependence (correlation) existing in the database can be leveraged by the adversary to infer more sensitive information thus violating privacy. Private attributes in a user’s record can be inferred by exploiting the public attributes of other users sharing similar interests \[17\]. A user’s susceptibility to a contagious disease can be easily inferred by an adversary who has access to noisy query results and is aware of the fact that the user’s immediate family members are part of the database being queried \[71\]. Social and behavioral dependence have also been used to perform de-anonymization attacks on released datasets \[64, 98, 103, 115\].

However, the state-of-the-art DP metric/mechanism does not explicitly account for dependence (correlation) across users’ records, leaving open the possibility of in-
ference attacks leveraging these correlations [71]. For example, in a social network graph (with nodes representing users, and edges representing ‘friendship’ relations), the ‘friendship’ between two nodes, not explicitly connected in the graph, can be inferred from the existence of edges between other nodes [82]. From the perspective of an individual user Alice, existing DP mechanisms ensure that Alice’s decision will not (significantly) influence the final outcome - any privacy harm that befalls Alice would have likely occurred even if she had made the choice to exclude her data from the dataset. In a game theoretic sense, this means that the action ‘include my data’ will be a dominant strategy for most individuals. However, when data is correlated, this equilibrium (where everyone participates) may not be globally optimal. In particular, differential privacy mechanisms do not imply that Alice will suffer no privacy harm due to the outcome of the mechanism. Furthermore, by choosing to include her data, Alice may negatively influence the privacy of other users in the dataset (e.g., Alice’s height and weight might be heavily correlated with her descendants). Therefore, instead of taking the perspective of an individual who chooses to include/exclude his data, we take the perspective of a data curator who already has collected a large dataset. The data curator thus ought to consider the impact of correlations in data while designing or applying data release mechanisms. This observation has motivated the development of recent works on Pufferfish privacy [71, 72], Blowfish privacy [57], Membership privacy [81] and Inferential privacy [49]. However, these approaches either (a) propose privacy metrics such as Pufferfish privacy and Membership privacy for which no general obfuscation mechanism is known [49, 71, 72, 81], or (b) do not fully incorporate probabilistic dependence between user records [57]. To overcome these challenges, we will propose our new framework of dependent differential privacy (DDP) in Chapter [3] that incorporates data dependence/correlation into the conventional framework of differential privacy. Our designed privacy mechanisms can strike
a balance between utility and privacy for dependent/correlated data in real world applications.

**How to quantify the state-of-the-art statistical privacy in an interpretable manner?**

While the concept of differential privacy has received considerable attention in the last decade, including industry and government adoption (e.g., Google, Apple, and US Census), there are very few guidelines on how to apply it in practice \[52,118\]. As illustrated by the recent controversy surrounding Apple’s implementation of differential privacy \[118\], a key challenge facing system designers and researchers is how to set appropriate values of the privacy parameters. Dwork and Smith have also identified this as an open research direction \[43\]. Specifically, they considered the choice of the privacy parameters as essentially a social question. However, existing tools provide only a limited support for understanding this social question. In addition, it has been observed in \[22,57,71,72,83,139\] that the appropriate choice of privacy parameters may also be affected by the existence of auxiliary information such as the static dependence/correlation or temporal dynamics in the database. To address these challenges, in Chapter 5, we will provide a rigorous and quantitative procedure to investigate the choice of an appropriate value of the privacy parameters, from the perspective of adversaries’ hypothesis testing. In Chapter 5, we also consider adversaries that have access to arbitrary auxiliary information such as the dependence (correlation) across records and the temporal dynamics of the database, especially focusing on their influence on the choice of privacy parameters.
1.2 Overview of the Dissertation

In this thesis, we aim to overcome the weaknesses of existing statistical privacy frameworks, in order to protect data privacy under temporal dynamics of network topologies and general data dependence/correlation, and provide useful and interpretable guidelines for selecting appropriate values of privacy parameters.

In Chapter 3, we propose a novel privacy-preserving system LinkMirage that mediates privacy-preserving access to users’ social relationships. LinkMirage takes users’ social relationship graph as an input, obfuscates the social graph topology, and provides untrusted external applications with an obfuscated view of the social relationship graph while preserving graph utility. Through theoretical analysis as well as using real-world social network topologies, we demonstrate the privacy and utility advantage of LinkMirage compared to the state-of-the-art approaches. LinkMirage can enable the design of real-world applications such as recommendation systems, graph analytics, anonymous communications, and Sybil defenses while protecting the privacy of social relationships.

In Chapter 4, we uniquely incorporate data correlation into differential privacy to propose the framework of dependent differential privacy (DDP). Furthermore, we propose a dependent perturbation mechanism (DPM) to achieve the privacy guarantees in DDP. Finally, using a combination of theoretical analysis and extensive experiments involving different classes of queries (e.g., machine learning queries, graph queries) issued over multiple large-scale real-world datasets, we show that our mechanism consistently outperforms the state-of-the-art approaches in managing the privacy-utility tradeoffs for dependent/correlated data.

In Chapter 5, we employ a statistical tool called hypothesis testing for discovering useful and interpretable guidelines for the state-of-the-art privacy-preserving frame-
works. We provide quantitative analysis of how hypothesis testing can guide the appropriate choice of the privacy parameter in an interpretable manner for a differentially private mechanism. We also consider adversaries that have access to arbitrary auxiliary information, especially focusing on their influence on the choice of the privacy parameters. In addition, we show how the perspective of hypothesis testing can provide useful insights on the relationships among a broad range of statistical privacy notions.

1.3 Dissertation Contributions

In summary, we make the following contributions to the area of statistical privacy:

Design of LinkMirage for Protecting Privacy under Temporal Dynamics of Network Topologies: In Chapter 3 we design LinkMirage to mediate privacy-preserving access to users’ social relationships. LinkMirage obfuscates links in the social graph (protecting link privacy) and provides untrusted external applications with an obfuscated view of the social graph. LinkMirage can achieve a good balance between privacy and utility, under the context of dynamic social network topologies.

LinkMirage provides rigorous privacy guarantees to defend against strategic adversaries with prior information of the social graph. We perform link privacy analysis both theoretically as well as using real-world social network topologies. The experimental results for both a Facebook dataset (with 870K links) and a large-scale Google+ dataset (with 940M links) show up to 10x improvement in privacy over the state-of-the-art research. We quantify a general utility metric for LinkMirage. We analyze our utility measurement provided by LinkMirage both theoretically and using real-world social graphs (Facebook and Google+). We experimentally demonstrate
the applicability of LinkMirage in real-world applications, such as privacy-preserving graph analytics, anonymous communication and Sybil defenses. LinkMirage enables the design of social relationships based systems while simultaneously protecting the privacy of users’ social relationships.

**Proposal of Dependent Differential Privacy Framework for Protecting Privacy under General Data Dependence/Correlation:** In Chapter 4, we formalize the notion of dependent differential privacy (DDP), to defend against adversaries who have prior information about the probabilistic dependence between tuples in a statistical database. We then show that it is possible to achieve the DDP guarantees by augmenting the Laplace mechanism, used for achieving the DP guarantees, with a dependence coefficient. The coefficient allows accurate computation of the query sensitivity for dependent data, thus minimizing the noise that needs to be added providing better utility at the same privacy level. Furthermore, we prove that our dependent perturbation mechanism is also resilient to composition attacks [33, 48].

Our proposed dependent perturbation mechanism applies to any class of query functions. Using extensive evaluation involving different query functions (e.g., machine learning queries such as clustering and classification, and graph queries such as degree distribution) over multiple large-scale real-world datasets we illustrate that our mechanism outperforms the state-of-the-art approaches in providing rigorous privacy and utility guarantees for dependent tuples.

**Quantification of Statistical Privacy and Guidelines for Selecting Appropriate Values of Privacy Parameters:** In Chapter 5, we investigate the framework of differential privacy from the perspective of the adversary’s hypothesis testing who observes differentially private outputs. Specifically, we theoretically analyze the capability of the adversary to infer mutually exclusive sensitive information about the input data (such as whether an individual has participated or not) from the output of
the privacy-preserving mechanism. We comprehensively analyze the (i) unbounded and (ii) bounded scenarios of differential privacy, and (iii) the relaxed variant of $(\epsilon, \delta)$-differential privacy. We quantify the success of the hypothesis testing using the precision-recall relation, which can serve as useful guidelines for selecting appropriate values of $\epsilon$ in an interpretable and quantitative manner.

Furthermore, we analyze the effect of three types of auxiliary information, namely, the prior distribution of the input record, the correlation across records, and the correlation across time, on the choice of the privacy parameter via the hypothesis testing by the adversary. We also systematically compare several state-of-the-art statistical privacy notions from the perspective of the adversary’s hypothesis testing, such as Pufferfish privacy [72], Blowfish privacy [57], dependent differential privacy [83], membership privacy [81], Inferential privacy [49] and mutual-information based differential privacy [26].

In summary, the dissertation systematically studies how to design effective statistical privacy frameworks and mechanisms for addressing practical privacy issues of protecting data under temporal dynamics of network topologies and general data dependence/correlation, and how to provide interpretable guidelines for selecting appropriate values of the privacy parameters.

1.4 Dissertation Organization

Chapter 1, this chapter, has introduced the motivation of this dissertation – information leakage is an increasing threat to data science, and thus statistical privacy has emerged as an important approach for protecting users’ sensitive information. This chapter also gives an overview of the dissertation.
Chapter 2 introduces the background necessary to understand the dissertation, as well as related work. We review statistical privacy in a broader sense and discuss various statistical privacy frameworks and mechanisms in detail.

Chapter 3 introduces the design of a privacy-preserving social relationship system named as LinkMirage to mediate users’ sensitive social relationships in both static and temporal scenarios while providing utility for general graph applications.

Chapter 4 introduces the framework of dependent differential privacy, that incorporates the dependence/correlation across users into the conventional framework of differential privacy. Our proposed mechanism can achieve a good privacy and utility trade-off under the dependent/correlated scenario.

Chapter 5 investigates the statistical privacy frameworks from the perspective of the adversary’s hypothesis testing. We provide a useful and interpretable guideline for selecting appropriate values of the privacy parameter, under different DP variants, adversaries with arbitrary auxiliary information and several state-of-the-art privacy frameworks.

Chapter 6 concludes this dissertation and summarizes opportunities for future work.
Chapter 2

Related Work

In this chapter, we first give a brief introduction to statistical privacy in the literature from a broad perspective. Then we focus on privacy-preserving graph data publishing, differential privacy and its variants, and applying hypothesis testing to analyze statistical privacy frameworks which are closely related to this dissertation.

Information sharing is key to realizing the vision of a data-driven customization of our environment. Data that were earlier locked up in private repositories are now being increasingly shared for enabling new context-aware applications, better monitoring of population statistics, and facilitating academic research in diverse fields. However, sharing personal data gives rise to serious privacy concerns as the data can contain sensitive information that a user might want to keep private. Thus, while on one hand, it is imperative to release utility-providing information, on the other hand, the privacy of users whose data is being shared also needs to be protected. In the literature, a number of techniques for data anonymization has been developed including $k$-anonymity [116], $l$-diversity [89], $t$-closeness [80] and differential privacy [34, 36, 39, 43], which have been widely accepted by the research community.
2.1 Privacy-preserving Graph Data Publishing

An increasing amount of data (such as social network data, business data, mobility data, medical data, etc.) is being generated by computer networks and services. This kind of data is referred to as structural data, and can typically be modeled as a graph. In a graph representation, the nodes represent entities (individuals or organizations), and the connections can represent friendships or interactions (for social network data), financial transactions (for business data), communication records (for mobility data), disease transmission (for medical data), etc. The sensitive nature of this data gives rise to significant security and privacy concerns.

To protect the privacy of sensitive data, two kinds of obfuscation properties are desirable: link privacy \[55, 56, 87, 95, 134, 137\], and vertex privacy \[9, 86, 106\]. In this thesis, we focus on obfuscating the structural properties of data via link privacy schemes. We note that techniques for link privacy can provide a strong foundation for vertex privacy. In the absence of such a foundation, the security community has been very successful in de-anonymizing the state-of-the-art perturbation approaches \[62, 64, 98, 103, 115\]. The sensitive links in structural data have many real-world applications. For instance, sensitive social trust relationships in social network data can be leveraged to improve security in anonymous communications and secure routing systems \[32, 96, 97\]. There exist fundamentally conflicting requirements for any data/link obfuscation mechanism: protecting privacy for the sensitive links in structural data and preserving utility of the obfuscated graph for use in real-world applications. On one extreme, we can protect privacy by publishing a random graph, but that does not preserve application utility. Therefore, our objective is to enable real-world applications while protecting the privacy of sensitive relationships (links) against undesirable inference attacks.
2.1.1 Graph Privacy with Labeled Vertices

An important thread of research aims to preserve link privacy between labeled vertices by obfuscating the edges, i.e., by adding/deleting edges [56, 87, 95, 134]. These methods aim to randomize the structure of the social graph, while differing in the manner of adding noise. Hay et al. [56] perturb the graph by applying a sequence of \( r \) edge deletions and \( r \) edge insertions. The deleted edges are uniformly selected from the existing edges in the original graph while the added edges are uniformly selected from the non-existing edges. However, neither the edge deletions nor edge insertions take any structural properties of the graph into consideration. Ying and Wu [134] proposed a new perturbation method for preserving spectral properties, without analyzing its privacy performance.

Mittal et al. proposed a perturbation method in [95], which serves as the foundation for our algorithm in Chapter 3. Their method deletes all edges in the original graph, and replaces each edge with a fake edge that is sampled based on the structural properties of the graph. In particular, random walks are performed on the original graph to sample fake edges. Let \( k \) denote the length of the random walk, which also serves as the perturbation parameter. The structure perturbation method is based on random walk on the original graph, where \( k \) is the random walk iteration number and controls the perturbation degree consequently. Consider each node \( u \) in the original graph \( G \) and its arbitrary neighbor \( v \). The algorithm considers a random walk of length \( k - 1 \) starting from \( v \), and samples the terminus point of the random walk \( w \). Then, a perturbed graph \( G' \) is constructed where the link \((u, w)\) is added (instead of the original link \((u, v)\)). Thus, the topology of the original graph is perturbed to provide link privacy. However, the structural properties can only be preserved under small perturbation parameter \( k \). Liu et al. further generalized this method by
automatically adapting the random walk length by using machine learning methods in [87].

Another line of research aims to preserve link privacy [55, 137] by aggregating the vertices and edges into super vertices. Therefore, the privacy of links within each super vertex is naturally protected. However, such approaches do not permit fine grained utilization of graph properties, making it difficult to be applied to applications such as social network based anonymous communication and Sybil defenses [27, 84, 135, 136].

2.1.2 Graph Privacy with Unlabeled Vertices

While the focus of our work is on preserving link privacy in context of labeled vertices, an orthogonal line of research aims to provide privacy in the context of unlabeled vertices (vertex privacy) [9, 86, 106]. Liu et al. [86] proposed $k$-anonymity to anonymize unlabeled vertices by placing at least $k$ vertices at an equivalent level. Differential privacy provides a theoretical framework for perturbing aggregate information, and Sala et al. [106] leveraged differential privacy to privately publish social graphs with unlabeled vertices. We note that LinkMirage proposed in Chapter 3 can also provide a foundation for preserving vertex privacy as stated in Section 3.6.2. Shokri et al. [111] addresses the privacy-utility trade-off by using game theory, which can be generalized to consider the temporal effects by updating the prior after each iteration. Furthermore, Theodorakopoulos et al. [120] take the correlation over time into account on the utility-privacy game designed in [113].

We further consider anonymity in temporal graphs with unlabeled vertices. The time series data should be seriously considered, since the adversaries can combine multiple published graph to launch enhanced attacks for inferring more information. Previous works explored privacy degradation in vertex privacy schemes due to the
release of multiple graph snapshots \[10, 30, 117\]. These observations motivate our work, even though we focus on labeled vertices.

**De-anonymization:** In recent years, the security community has proposed a number of sophisticated attacks for de-anonymizing social graphs \[14, 62, 64, 98, 103, 115\]. While most of these attacks are not applicable to link privacy mechanisms (their focus is on vertex privacy), they illustrate the importance of considering adversaries with prior information about the social graph. Narayanan et al. \[98\] effectively de-anonymized a Twitter dataset by utilizing a Flickr dataset as auxiliary information. Furthermore, Nilizadeh et al. \[103\] exploited the community structure of a graph to de-anonymize social networks. Other public datasets may also contain individual behavior information. For instance, Srivatsa et al. \[115\] proposed to de-anonymize a set of location traces based on a social network. They demonstrated that a contact graph identifying meetings between anonymized users in the location traces can be structurally correlated with the corresponding social network graph. Burattin et al. \[14\] exploited inadvertent information leaks via Facebook’s graph API to de-anonymize social links; Facebook’s new graph API (v2.0) features stringent privacy controls as a countermeasure. We perform a rigorous privacy analysis of LinkMirage (Section 3.5) by considering a worst-case (strongest) adversary that knows the entire social graph except one link, and show that even such an adversary is limited in its inference capability.

### 2.1.3 Preserving Privacy for Social Networks

In Chapter \[3\] we consider social networks as an important representative example of graph data. Online social networks (OSNs) have revolutionized the way our society interacts and communicates with each other. Under the hood, OSNs can be viewed as
a special graph structure composed of individuals (or organizations) and connections between these entities. These social relationships represent sensitive relationships between entities, for example, trusted friendships or important interactions in Facebook, Twitter, or Google+, which users want to preserve the security and privacy of.

At the same time, an increasing number of third party applications rely on users’ social relationships (these applications can be external to the OSN). E-commerce applications can leverage social relationships for improving sales [73], and data-mining researchers also rely on the social relationships for functional analysis [100,104]. Social relationships can be used to mitigate spam [94]. Anonymous communication systems can improve client anonymity by leveraging users’ social relationships [32,96,97]. State-of-the-art Sybil defenses rely on social trust relationships to detect attackers [27,135].

However, both users and the OSN providers are hesitant to share social relationships/graphs with these applications due to privacy concerns. For instance, a majority of users are exercising privacy controls provided by popular OSNs such as Facebook, Google+ and LinkedIn to limit access to their social relationships [29]. Privacy concerns arise because external applications that rely on users’ social relationships can either explicitly reveal this information to an adversary, or allow the adversary to perform inference attacks [50,64,82,98,103,115]. These concerns hinder the deployment of many real-world applications. Thus, there exist fundamentally conflicting requirements for any link obfuscation mechanism: protecting privacy for the sensitive links in social networks and preserving utility of the obfuscated graph for use in real-world applications. Ji et al. [63] analyze and systematize the state-of-the-art graph data privacy and utility techniques.

In Chapter 3, we design LinkMirage, a system that mediates privacy-preserving access to social relationships. LinkMirage takes users’ social relationship graph (can
accommodate both static and temporal scenarios) as an input, either via an OSN operator or via individual user subscriptions. Next, LinkMirage *obfuscates* the social graph topology to protect the privacy of users’ social contacts (edge/link privacy, not vertex privacy). LinkMirage then provides external applications such as graph analytics and anonymity systems with an obfuscated view of the social relationship graph. Thus, LinkMirage provides a trade-off between securing the confidentiality of social relationships, and enabling the design of social relationship based applications.

### 2.2 Differential Privacy Frameworks

Differential privacy (DP) is one of the most popular statistical privacy frameworks. Query answering algorithms that satisfy differential privacy produce noisy query answers such that the distribution of the query answers changes only slightly with the addition, deletion or modification of any tuple. The threat to privacy arises from the release of aggregate query results computed over the statistical database. The goal of differential privacy is to randomize the query results to ensure that the risk to the user’s privacy does not increase substantially (bounded by a function of $\epsilon$) as a result of participating in the statistical database.

We represent a statistical database using a vector $D = [D_1, D_2, \cdots, D_n]$ drawn from a distribution $\mathcal{D}$, where $D_i \in \mathcal{R}^m$ denotes the data of the $i^{th}$ user. Let $\mathcal{A}(D)$ be the output of a randomized algorithm $\mathcal{A}$ applied to database $D$. The notion of $\epsilon$-differential privacy is formally defined as:

**Definition 1.** ($\epsilon$-differential privacy) A randomized algorithm $\mathcal{A}$ provides $\epsilon$-differential privacy if for any two neighboring databases $D$ and $D'$ such that $D$ and
D' differ by adding/removing a record, and for any output S,

\[
\max_{D,D'} \frac{P(A(D) \in S)}{P(A(D') \in S)} \leq \exp(\epsilon)
\]

(2.1)

where \(A(D)\) (resp.\(A(D')\)) is the output of \(A\) on input \(D\) (resp.\(D'\)) and \(\epsilon\) is the privacy budget.

2.2.1 Achieving Differential Privacy

The Laplace Perturbation Mechanism (LPM) is a popular framework to achieve \(\epsilon\)-differential privacy [34]. The key idea is to use the noise drawn from a suitable Laplace distribution to perturb the query results before their release. Let \(Lap(\sigma)\) denote a zero mean Laplace distribution with scaling factor \(\sigma\). The corresponding density function is given by \(f(x) = \frac{1}{2\sigma} \exp\left(-\frac{|x|}{\sigma}\right)\). For a query output of dimension \(q\), we use a noise vector \(Lap^q(\sigma)\) where each dimension of the vector is generated independently according to \(Lap(\sigma)\).

A fundamental element in the design of the LPM is the global sensitivity of the aggregate query. Let \(Q : D \rightarrow \mathbb{R}^q\) be a query function issued on database \(D\). The global sensitivity \(\Delta Q\) of \(Q\) is defined as:

**Definition 2.** (Global sensitivity) [34] The global sensitivity of a query function \(Q : D \rightarrow \mathbb{R}^q\) is the maximum difference between the values of the function when one input changes. Formally,

\[
\Delta Q = \max_{D,D'} \|Q(D) - Q(D')\|_1
\]

(2.2)

**Theorem 1.** \(\epsilon\)-Differential privacy is guaranteed if the scaling factor \(\sigma\) in the Laplace distribution is calibrated according to the global sensitivity \(\Delta Q\) [34]. For any query
function $Q$ over an arbitrary domain $D$, the mechanism $A$

$$
A(D) = Q(D) + \text{Lap}(\Delta Q/\epsilon)
$$

achieves $\epsilon$-differential privacy.

### 2.2.2 Categories of Differential Privacy

Definition 1 is also known as unbounded differential privacy as the database size is unknown. When the database size is known, $D$ and $D'$ are neighbors if $D$ can be obtained from $D'$ by replacing one record in $D'$ with another record. Definition 1 based on this notion of neighbors is known as bounded differential privacy [71]. Differential privacy also provides rigorous privacy guarantees under multiple queries. The privacy property degrades linearly under sequential queries; for parallel queries, the degraded privacy guarantee is the worst of the guarantees offered by the individual releases [48, 67, 90]. Specifically, differential privacy satisfies both sequential composition theorem and parallel composition theorem as follows.

**Theorem 1. (Sequential Composition Theorem [90])** Let $A_t(\cdot)$ each provide $\epsilon_t$-differential privacy. A sequence of $A_t(\cdot)$ computed over the database $D$ provides $\sum_t \epsilon_t$-differential privacy.

**Theorem 2. (Parallel Composition Theorem [90])** Let $A_t(\cdot)$ each provide $\epsilon_t$-differential privacy. A sequence of $A_t(\cdot)$ computed over a disjoint subset of database $D_t$ provides $\max_t \epsilon_t$-differential privacy.
Approximate differential privacy is another variant of differential privacy, also named \((\epsilon, \delta)\)-differential privacy \cite{37}, and is defined as

\[
P(\mathcal{A}(D) \in S) \leq P(\mathcal{A}(D') \in S) \exp(\epsilon) + \delta \tag{2.4}
\]

for any two neighboring databases \(D\) and \(D'\). Approximate differential privacy relaxes the conventional differential privacy by ignoring noisy outputs with a certain probability controlled by the parameter \(\delta\).

### 2.2.3 Implementation and Analysis of Differential Privacy

According to the US chief scientific officer, the US Census in 2020 is planning to have all its datasets differentially private for protecting specific sensitive information. Google has explored differential privacy with its randomized aggregatable privacy-preserving ordinal response (RAPPOR) work \cite{44} that has been tested in Chrome. With differential privacy, Apple can collect and store its users’ data in a format that lets it glean useful notions about what people do without extracting anything about a single, specific one of those people. Johnson, Near and Song \cite{65} propose a practical end-to-end system to enforce differential privacy for SQL queries, which has recently been adopted by Uber to enforce differential privacy for internal data analytics \cite{2}.

Kifer and Machanavajjhala \cite{71} argue that differential privacy cannot guarantee privacy and utility without stating how the data are generated (the assumptions about the adversaries). Cuff and Yu \cite{26} discuss the relation between the conditional mutual information and differential privacy from an information theoretic perspective. Kasiviswanathan and Smith \cite{68} provide a Bayesian formulation of differential privacy. Haeberlen et al. \cite{52} investigate the vulnerabilities of differential privacy under side-channel attacks.
2.2.4 Differential Privacy under Adversaries’ Auxiliary Information

Tuples in real-world data often exhibit inherent dependence or correlations. Handling dependent tuples is a significant problem. Kifer and Machanavajjhala [71] were the first to argue that the dependence (correlation) existing among tuples may degrade the expected privacy levels of an individual in existing DP mechanisms. In contrast to their theoretical work, we demonstrate inference attacks using real data on complex differentially private machine learning queries (Chapter 4). Our inference attack demonstrates that the auxiliary information would also be useful to infer an individual’s information from noisy query results generated by existing differential privacy mechanisms.

Kifer et al. proposed the Pufferfish privacy framework [72] to provide rigorous privacy guarantees against adversaries who may have access to any auxiliary background information and side information of the database. Blowfish privacy [57] is a subclass of Pufferfish which only considered the data correlations introduced by the deterministic constraints. Our proposed dependent differential privacy in Chapter 4 is highly motivated by these privacy frameworks and is a subclass of the Pufferfish framework that takes the probabilistic dependence relationships into consideration. In Chapter 4, we further propose our dependent perturbation mechanism to rigorously achieve dependent differential privacy for general query functions. Membership privacy [81] is also applicable for dependent data, however limited anonymization algorithms have been proposed for this framework. Chen et al. [22] dealt with the correlated data by multiplying the original sensitivity with the number of correlated records. We have shown, both theoretically and experimentally, that the baseline approach would introduce a large amount of noise and thus deteriorate the utility performance of query answers. Zhu et al. [139] exploited the linear relationships among tuples which does
not satisfy any rigorous privacy metric. Furthermore, their method has been verified to have significantly worse privacy and utility performance compared to our DPM.

Tschantz, Sen and Datta [121] investigate most of the previous work that study differential privacy under correlated data, and classify them under two perspectives named as causal and associative views respectively. The perspective of causal model considers an intervention which artificially alters input as did in experiments as opposed to naturally. It is interesting to note that the causal perspective requires to distinguish an individual’s attributes (‘secrets about you’) and the data that is the input to the algorithm (‘secrets from you’). Therefore, the framework of differential privacy has already taken this causal perspective into consideration. In comparison, the associative perspective aims to explicitly model the dependence/correlation relationship naturally existing in the database, which motivates the proposal of our dependent differential privacy frameworks.

Furthermore, continuously generated data in the real world tend to be temporally correlated, and such correlations can also be acquired by adversaries. Xiao et al. [132] proposed to exploit temporal correlation in a single user scenario in the location-based applications. Cao et al. [16] investigate the potential privacy loss of a traditional Differential privacy mechanism under temporal correlations in the context of continuous data release. Other privacy mechanisms [45,66,110] have been proposed to privately learn desired statistics over multiple users’ time-series data. However, these works [16,45,66,110,132] only considered the correlation among the same user’s data in different timestamps, which is different from our settings in Chapter 4 where different users’ data are correlated.

In Chapter 5, we will investigate adversaries that have access to arbitrary auxiliary information — in the form of prior distribution of the database and correlation
across records and time, especially focusing on their influence on the choice of privacy parameters.

### 2.3 Hypothesis Testing and Differential Privacy

A key issue in deploying differential privacy and other statistical privacy frameworks is the selection of proper values of the privacy parameters. In Chapter 5, we aim to leverage the techniques of hypothesis testing to develop an effective guideline for selecting appropriate values of the privacy parameters.

#### 2.3.1 Hypothesis Testing

Hypothesis testing \[^{[5,99,129]}\] is the use of statistics on the observed data to determine the probability that a given hypothesis is true. The common process of hypothesis testing consists of four steps.

- Step 1: State the hypotheses;
- Step 2: Set the criterion for a decision;
- Step 3: Compute the test statistic;
- Step 4: Make a decision.

The binary hypothesis testing problem[^{[5,99]}] decides between a null hypothesis $H = h_0$ and an alternative hypothesis $H = h_1$ based on observation of a random variable $O$. Under hypothesis $h_0$, $O$ follows the probability distribution $P_0$, while

---

[^1]: We consider the binary hypothesis testing problem since the adversary aims to distinguish two neighboring databases in differential privacy.
under $h_1$, $O$ follows distribution $P_1$. A decision rule $\hat{H}$ is a criterion that maps every possible observation $O = o$ to either $h_0$ or $h_1$.

The most popularly used criteria for decision are maximum likelihood \cite{109}, maximum posterior probability \cite{51}, minimum cost \cite{92}, and Neyman-Pearson criterion \cite{102}.

**Definition 3.** The Neyman-Pearson criterion aims to maximize the true detection rate (the probability of correctly accepting the alternative hypothesis) subject to a maximum false alarm rate (the probability of mistakenly accepting the alternative hypothesis) \cite{102,129}, i.e.,

$$\max P_{TD} \text{ s.t., } P_{FA} \leq \alpha$$

where $P_{TD}$ and $P_{FA}$ denote the true detection rate and the false alarm rate, respectively.

The Neyman-Pearson criterion has the highest statistical power \cite{79} since it maximizes the true detection rate under a given requirement of the false alarm rate (as defined above). According to the Neyman-Pearson Lemma \cite{101}, an efficient way to solve Eq. 2.5 is to use the likelihood ratio test \cite{107,125}. In practice, the likelihood ratio, or equivalently its logarithm, can be used directly to construct test statistics to compare the goodness of fit of the two hypotheses. Other criteria such as maximum likelihood \cite{109}, maximum posterior probability \cite{51} and minimum cost \cite{92} cannot guarantee the highest statistical power in general, and they are based on a fixed threshold on the likelihood ratio that cannot incorporate $\alpha$. In contrast, the Neyman-Pearson criterion treats the testing performance as a function of the threshold for the likelihood ratio controlled by $\alpha$ \cite[pp.67]{1}. Therefore, the Neyman-Pearson criterion is the most powerful test which also provides more flexibility to optimize a range of evaluation metrics by setting different values of $\alpha$ \cite{79}. 
2.3.2 Hypothesis Testing in Differential Privacy

Previous work in [47, 105, 126] has investigated how to accurately compute the test statistics in hypothesis testing while using differential privacy to protect data, which is orthogonal to our setting. Ding et al. designed an algorithm to detect privacy violations of DP mechanisms from a hypothesis testing perspective [31]. Wasserman and Zhou [128], Hall et al. [53], and Kairouz et al. [67] are our inspiration for using hypothesis testing on differential privacy. These works propose a view of differential privacy from the perspective of hypothesis testing by the adversary who observes the differentially private outputs. Specifically, the observation is first made by Wasserman and Zhou [128] for bounding the probability for the adversary to correctly reject a false hypothesis of the input database. Hall et al. [53] later extend such analysis to $(\epsilon, \delta)$-differential privacy. Kairouz et al. [67] then apply this concept in their proof of composition of differential privacy. However, these prior works have not applied hypothesis testing to our objectives of determining the appropriate value of $\epsilon$.

In contrast, our work in Chapter 5 extensively analyzes the adversary’s capability of implementing hypothesis testing using Neyman-Pearson’s criterion [102] on outputs of differential privacy mechanisms, such as LPM and the Gaussian perturbation mechanism. We apply it to the problem of determining $\epsilon$, the analysis of the effect of auxiliary information on the choice of $\epsilon$, and the comparative analysis of the privacy guarantee provided by different privacy frameworks. To our knowledge, we are the first to comprehensively investigate how hypothesis testing can be used as a viable tool to guide the selection of the privacy parameter in the design of privacy preservation frameworks. Finally, we note that our approach is not limited to the aforementioned settings and can be generalized to other hypothesis testing techniques and other perturbation mechanisms as well.
2.3.3 Setting the Privacy Budget

Setting the privacy budget $\epsilon$ in DP is a challenging task. Prior work [61,74,76,77] attempted to address this problem, but has several limitations. Hsu et al. [61] proposed an economic method to express the balance between the accuracy of a DP release and the strength of privacy guarantee in terms of a cost function when bad events happen. However, this work involves complicated economic models consisting of bad events and their corresponding cost functions. It is difficult to quantify the cost of a bad event for general applications. Krehbiel [74] takes each data owner’s privacy preference into consideration and aims to select a proper privacy parameter for achieving a good tradeoff between utility and privacy. This mechanism focuses more on an economic perspective for collecting and distributing payments under a chosen level of privacy parameter and their privacy definition is not the standard DP.

Other related works by Lee and Clifton [76,77] determine $\epsilon$ for LPM based on the posterior probability that the adversary can infer the value of a record. The Neyman-Pearson criterion adopted in our approach (Chapter 5) has an advantage over the maximum posterior probability analysis [51]. As stated in Section 2.3, the Neyman-Pearson criterion in our work can be used to optimize a range of evaluation metrics by selecting a proper value of the false alarm rate while maximizing the true detection rate. In addition, their analysis [76] only assumes uniform distribution of the input data (equivalent to the scenarios without any prior distribution) in both their experiments and theoretical derivations.

Finally, these previous works are noticeably different from our approach in Chapter 5 as they do not utilize the statistical tool of hypothesis testing (especially the Neyman-Pearson criterion) by the adversary. Furthermore, our work is not limited to the determination of the $\epsilon$ value, but is also extended to the analysis on the impact of
the auxiliary information possessed by the adversary and the comparison across the state-of-the-art statistical privacy frameworks, which has not been studied before.
Chapter 3

LinkMirage: Enabling Privacy-preserving Analytics on Correlated and Evolving Graph Data

We consider social networks as an important representative example of graph data. Social relationships present a critical foundation for many real-world applications. However, both users and online social network (OSN) providers are hesitant to share social relationships with untrusted external applications due to privacy concerns. In this work, we design LinkMirage, a system that mediates privacy-preserving access to social relationships. LinkMirage takes users’ social relationship graph (can accommodate correlated and evolving graph data) as an input, obfuscates the social graph topology, and provides untrusted external applications with an obfuscated view of the social relationship graph while preserving graph utility.

Our key contributions are (1) a novel algorithm for obfuscating social relation-
ship graph while preserving graph utility, (2) theoretical and experimental analysis of privacy and utility using real-world social network topologies, including a large-scale Google+ dataset with 940 million links. Our experimental results demonstrate that LinkMirage provides up to 10x improvement in privacy guarantees compared to the state-of-the-art approaches. Overall, LinkMirage enables the design of real-world applications such as recommendation systems, graph analytics, anonymous communications, and Sybil defenses while protecting the privacy of social relationships.

3.1 Motivation

In this work, we focus our research on protecting the link privacy between labeled vertices in social networks. Mechanisms for graph analytics, anonymous communication, and Sybil defenses can leverage users’ social relationships for enhancing security, but end up revealing users’ social relationships to adversaries. For example, in the Tor network, the relays’ IP addresses (labels) are already publicly known (vertex privacy in is not useful). Tor operators are hesitant to utilize social trusts to set up the Tor circuit as recommended by since the circuit construction protocol would reveal sensitive social contact information about the users. Our proposed link-privacy techniques can thus be utilized by the Tor relay operators to enhance system security while preserving link privacy. Overall, our work focuses on protecting users’ trust relationships while enabling the design of such systems.

LinkMirage supports three categories of social relationship based applications: 1) Global access to the obfuscated graph: Applications such as social network based anonymity systems and peer-to-peer networks can utilize LinkMirage

1The materials in this chapter is based on [85].
(described in Section 3.4.2) to obtain a global view of privacy-preserving social graph topologies; 2) Local access to the obfuscated graph: an individual user can query LinkMirage for his/her obfuscated social relationships (local neighborhood information), to facilitate distributed applications such as SybilLimit [135]; 3) Mediated data analytics: LinkMirage can enable privacy-preserving data analytics by running desired functional queries (such as computing graph modularity and pagerank score) on the obfuscated graph topology and only returning the result of the query, since immediate analytics operated on the original graph topology would leak sensitive information about users’ social relationships. Existing work [34, 38] demonstrated that the implementation of graph analytics algorithms could leak certain information. Instead of repeatedly adding perturbations to the output of each graph analytics algorithm as in differential privacy [34, 38], which would be rather costly, LinkMirage can obtain the perturbed graph just once to support multiple graph analytics. Such an approach protects the privacy of users’ social relationships from inference attacks using query results. There exists a plethora of attacks against vertex anonymity based mechanisms [64, 98, 103, 115]. Ji et al. [63] recently showed that no single vertex anonymization technique was able to resist all the existing attacks. Note that these attacks are not applicable to link privacy schemes. Therefore, a sound approach to vertex anonymity must start with improvements in our understanding of link privacy. When used as first step in the design of vertex privacy mechanisms, our approach can protect the privacy of social contacts and graph links even when the vertices are de-anonymized using state-of-the-art approaches [64, 98, 103, 115]. Furthermore, our method can even improve the resilience of vertex anonymity mechanisms against de-anonymization attacks when applied to unlabelled graphs (will be shown in Section 3.6.2).

We present a novel obfuscation algorithm that first clusters social graphs, and then anonymizes intra-cluster links and inter-cluster links, respectively. We obfuscate links
in a manner that preserves the key structural properties of social graphs. While our approach is of interest even for static social graphs, we go a step further in this work, and consider the evolutionary dynamics of social graphs (node/link addition or deletion). We design LinkMirage to be resilient to such evolutionary dynamics, by consistently clustering social graphs across time instances. Consistent clustering improves both the privacy and utility of the obfuscated graphs. We show that LinkMirage provides strong privacy properties. Even a strategic adversary with full access to the obfuscated graph and prior information about the original social graph is limited in its ability to infer information about users’ social relationships. LinkMirage provides up to 3x privacy improvement in static settings, and up to 10x privacy improvement in dynamic settings compared to the state-of-the-art approaches. Overall, our work makes the following contributions.

- First, we design LinkMirage to mediate privacy-preserving access to users’ social relationships. LinkMirage obfuscates links in the social graph (link privacy) and provides untrusted external applications with an obfuscated view of the social graph. LinkMirage can achieve a good balance between privacy and utility, under the context of both static and dynamic social network topologies.

- Second, LinkMirage provides rigorous privacy guarantees to defend against strategic adversaries with prior information of the social graph. We perform link privacy analysis both theoretically as well as using real-world social network topologies. The experimental results for both a Facebook dataset (with 870K links) and a large-scale Google+ dataset (with 940M links) show up to 10x improvement in privacy over the state-of-the-art research.

- Third, we experimentally demonstrate the applicability of LinkMirage in real-world applications, such as privacy-preserving graph analytics, anonymous communication and Sybil defenses. LinkMirage enables the design of social rela-
tionships based systems while simultaneously protecting the privacy of users’ social relationships.

• Finally, we quantify a general utility metric for LinkMirage. We analyze our utility measurement provided by LinkMirage both theoretically and using real-world social graphs (Facebook and Google+).

The frequently used mathematical notations in this chapter are listed in Table 3.1.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G_t$</td>
<td>The $t$-th original graph</td>
</tr>
<tr>
<td>$V_t$</td>
<td>The set of nodes in the $t$-th original graph</td>
</tr>
<tr>
<td>$E_t$</td>
<td>The set of nodes in the $t$-th original graph</td>
</tr>
<tr>
<td>$G'_t$</td>
<td>The $t$-th perturbed graph</td>
</tr>
<tr>
<td>$C_t$</td>
<td>The clusters of the $t$-th original graph</td>
</tr>
<tr>
<td>$C'_t$</td>
<td>The clusters of the $t$-th perturbed graph</td>
</tr>
<tr>
<td>$ch, un, in$</td>
<td>Short for changed, unchanged, inter-community</td>
</tr>
<tr>
<td>$k$</td>
<td>The length of random walk in the perturbation mechanism</td>
</tr>
<tr>
<td>$l$</td>
<td>The parameter of application level</td>
</tr>
<tr>
<td>$</td>
<td>\cdot</td>
</tr>
<tr>
<td>$\text{deg}(\cdot)$</td>
<td>The degree of a node</td>
</tr>
</tbody>
</table>
3.2 System Architecture and Threat Model

Fig. 3.1 shows the overall architecture for LinkMirage. For link privacy, we consider the third-party applications (which can query the social link information) as adversaries, which aim to obtain sensitive link information from the perturbed query results. A sophisticated adversary may have access to certain prior information such as partial link information of the original social networks, and such prior information can be extracted from publicly available sources, social networks such as Facebook, or other application-related sources as stated in [14]. The adversary may leverage Bayesian inference to infer the probability for the existence of a link. We assume that LinkMirage itself is trusted, in addition to the social network providers/users who provide the input social graph. LinkMirage first collects social link information through our social link app or directly through the OSN providers, and then applies an obfuscation algorithm to perturb the original social graph(s). The obfuscated graph(s) would be utilized to answer the query of the untrusted applications in a privacy-preserving manner.

Figure 3.1. **LinkMirage architecture.** LinkMirage first collects social link information through our social link app or directly through the OSN providers, and then applies an obfuscation algorithm to perturb the original social graph(s). The obfuscated graph(s) would be utilized to answer the query of the untrusted applications in a privacy-preserving manner. The third-party application (which queries the social link information) is considered an adversary which aims to obtain sensitive link information from the perturbed query results.
In Section 3.5.2 [3.5.3] we define our Bayesian privacy metric (called *anti-inference privacy*) and an information theoretic metric (called *indistinguishability*) to characterize the privacy offered by LinkMirage against adversaries with prior information. In addition, the evolving social topologies introduce another serious threat where sophisticated adversaries can combine information available in multiple query results to infer users’ social relationships. We define *anti-aggregation privacy* in Section 3.5.4 for evaluating the privacy performance of LinkMirage against such adversaries.

**Basic Theory:** Let us denote a time series of social graphs as $G_0, \cdots, G_T$. For each temporal graph $G_t = (V_t, E_t)$, the set of vertices is $V_t$ and the set of edges is $E_t$. For our theoretical analysis, we focus on undirected graphs where all the $|E_t|$ edges are symmetric, i.e. $(i, j) \in E_t$ iff $(j, i) \in E_t$. Note that our approach can be generalized to directed graphs with asymmetric edges. $P_t$ is the transition probability matrix of the Markov chain on the vertices of $G_t$. $P_t$ measures the probability that we follow an edge from one vertex to another vertex, where $P_t(i, j) = 1/\text{deg}(i)$ ($\text{deg}(i)$ denotes the degree of vertex $i$) if $(i, j) \in E_t$, otherwise $P_t(i, j) = 0$. A random walk starting from vertex $v$, selects a neighbor of $v$ at random according to $P_t$ and repeats the process.

### 3.3 System Overview and Roadmap

#### 3.3.1 LinkMirage System

Our objective for LinkMirage is to obfuscate social relationships to strike a balance between privacy for users’ social relationships and the usability for large-scale real-world applications, as will be discussed in Section 3.4.1. We deploy LinkMirage as a Facebook application that implements graph construction and obfuscation, as will be
discussed in Section 3.4.2.

For our perturbation mechanism of LinkMirage, we take both the static and the temporal social network topology into consideration, as will be discussed in Section 3.4.3. Our obfuscation mechanism consists of the following conceptual steps:

- Dynamic clustering which finds community structures in evolving graphs by simultaneously considering consecutive graphs. Our dynamic clustering utilizes an effective backtracking strategy to cluster a graph based on the clustering result of the previous graph.

- Selective perturbation which perturbs the minimal amount of edges in the evolving graphs according to the dynamic clustering result, where we only perturb the changed communities between consecutive graphs. In this manner, it is possible to use a very high privacy parameter in the perturbation process, while preserving structural properties of the social network topologies.

We further discuss the scalability of our perturbation algorithm on the real world large-scale Google+ dataset in Section 3.4.4 and visually show the effectiveness of our algorithm on the real world Facebook dataset in Section 3.4.5.

3.3.2 Privacy Evaluation

In Section 3.5, we rigorously analyze the privacy advantage of our LinkMirage over the state-of-the-art approaches, by considering three adversarial scenarios where sophisticated adversaries can combine information available in multiple query results to infer users’ social relationships. The privacy advantage of our LinkMirage will be demonstrated as
• LinkMirage shows significant privacy advantages in anti-inference privacy (will be defined in Section 3.5.2). The difference between the posterior probability and the prior probability of the existence of an link is smaller for LinkMirage than the state-of-the-art methods.

• LinkMirage achieves higher indistinguishability from an information theoretic perspective (will be defined in Section 3.5.3) where the obfuscated graph of LinkMirage contains less information of users’ social relationships than the state-of-the-art methods.

• LinkMirage also shows significant privacy advantages in anti-aggregation privacy (will be defined in Section 3.5.4), where the adversary’s estimation for users’ social relationships is less accurate for LinkMirage than the existing methods.

3.3.3 Utility Evaluation

In Section 3.6 we apply our perturbation algorithm to various real world applications such as graph analytics, anonymous communications, and Sybil defenses. Compared to previous methods, LinkMirage results in significantly lower attack probabilities when applied to anonymous communications and higher resilience to de-anonymization attacks when applied to vertex anonymity systems. LinkMirage even surprisingly improves the Sybil detection performance when applied to the distributed SybilLimit systems. LinkMirage also outperform existing methods in preserving the utility for multiple graph analytics applications, such as pagerank score and modularity. In Section 3.7 we further analyze LinkMirage’s ability to preserve general graph-theoretic characteristics.
3.4 LinkMirage System

3.4.1 Design Goals

We envision that applications relying on social relationships between users can bootstrap this information from online social network operators such as Facebook, Google+, Twitter with access to the users’ social relationships. To enable these applications in a privacy-preserving manner, a perturbed social graph topology (by adding noise to the original graph topology) should be available.

Social graphs evolve over time, and the third-party applications would benefit from access to the most current version of the graph. A baseline approach is to perturb each graph snapshot independently. However, the sequence of perturbed graphs provide significantly more observations to an adversary than just a single perturbed graph. We argue that an effective perturbation method should consider the evolution of the original graph sequence. Therefore, we have the overall design goals for our system as:

1. We aim to obfuscate social relationships while balancing privacy for users’ social relationships and the usability for real-world applications.

2. We aim to handle both the static and dynamic social network topologies.

3. Our system should provide rigorous privacy guarantees to defend against adversaries who have prior information of the original graphs, and adversaries who can combine multiple released graphs to infer more information.

4. Our method should be scalable to be applied in real-world large-scale social graphs.
3.4.2 LinkMirage: Deployment

To improve the usability of our proposed obfuscation approach (which will be described in detail in Section 3.4.3), and to avoid dependance on the OSN providers, we developed a Facebook application that implements graph construction (via individual user subscriptions) and obfuscation. The work flow of the LinkMirage deployment is as follows: (i) When a user visits the above URL, Facebook checks the credentials of the user, asks whether to grant the user’s friends permission, and then gets redirected to the application hosting server. (ii) The application server authenticates itself, and then queries Facebook for the information of the user’s friends, and returns their information such as user’s id. The list of user’s friends can then be collected by the application server to construct a Facebook social graph for the current timestamp. Leveraging LinkMirage, a perturbed graph for this timestamp would be available which preserves the link privacy of the users’ social relationships.

Real-world systems such as Uproxy, Lantern, Kaleidoscope [59], anonymity systems [32, 96, 97], Sybil defenses systems [27, 135] can directly benefit from our protocol through automatically obtaining the perturbed social relationships. Furthermore, our protocol can enable privacy-preserving graph analytics for OSN providers.

3.4.3 LinkMirage: Perturbation Algorithm

Social networks evolve with time and publishing a time series of perturbed graphs raises a serious privacy challenge: an adversary can combine information available from multiple perturbed graphs over time to compromise the privacy of users’ social contacts [10, 30, 117]. In LinkMirage, we take a time series of graph topologies into consideration, to account for the evolution of the social networks. Intuitively, the scenario with a static graph topology is just a special situation of the temporal graph
Consider a social graph series $G_0 = (V_0, E_0), \cdots, G_T = (V_T, E_T)$. We want to transform the graph series to $G'_0 = (V'_0, E'_0), \cdots, G'_T = (V'_T, E'_T)$, such that the vertices in $G'_t$ remain the same as in the original graph $G_t$, but the edges are perturbed to protect link privacy. Moreover, while perturbing the current graph $G_t$, LinkMirage has access to the past graphs in the time series (i.e., $G_0, \cdots, G_{t-1}$). Our perturbation goal is to balance the utility of social graph topologies and the privacy of users’ social contacts, across time.

**Figure 3.2. Our perturbation mechanism for $G_t$.** Assume that $G_{t-1}$ has already been dynamically obfuscated, based on dynamic clustering (step 1) and selective perturbation (step 2). Our mechanism analyzes the evolved graph $G_t$ (step 3) and dynamically clusters $G_t$ (step 4) based on the freed $m$ hop neighborhood ($m = 2$) of new links (between green and blue nodes), the merging virtual node (the large red node in step 4), and the new nodes. By comparing the communities in $G_{t-1}$ and $G_t$, we can implement selective perturbation (step 5), i.e. perturb the changed blue community independently and perturb the unchanged red and green communities in the same way as $G'_{t-1}$, and then perturb the inter-cluster links.

*Approach Overview:* Our perturbation mechanism for LinkMirage is illustrated in Fig. 3.2.
**Static scenario:** For a static graph \( G_{t-1} \), we first cluster it into several communities, and then perturb the links within each community. The inter-cluster links are also perturbed to protect their privacy.

**Dynamic scenario:** Let us suppose that \( G_t \) evolves from \( G_{t-1} \) by addition of new vertices (shown in blue color). To perturb graph \( G_t \), our intuition is to consider the similarity between graphs \( G_{t-1} \) and \( G_t \).

First, we partition \( G_{t-1} \) and \( G_t \) into subgraphs, by clustering each graph into different communities. To avoid randomness (guarantee consistency) in the clustering procedure and to reduce the computation complexity, we dynamically cluster the two graphs *together* instead of clustering them independently. Noting that one green node evolves by connecting with a new blue node, we free all the nodes located within \( m = 2 \) hops of this green node (the other two green nodes and one red node) and merge the remaining three red nodes to a big virtual node. Note that we free the nodes from the previously clustering hierarchy. Then, we cluster these new nodes, the freed nodes and the remaining virtual node to detect communities in \( G_t \).

Next, we compare the communities within \( G_{t-1} \) and \( G_t \), and identify the *changed* and *unchanged* subgraphs. For the *unchanged* subgraphs \( C_1, C_2 \), we set their perturbation at time \( t \) to be identical to their perturbation at time \( t - 1 \), denoted by \( C'_1, C'_2 \). For the *changed* subgraph \( C_3 \), we perturb it independently to obtain \( C'_3 \). We also perturb the links between communities to protect privacy of these inter-cluster links. Finally, we publish \( G'_t \) as the combination of \( C'_1, C'_2, C'_3 \) and the perturbed inter-cluster links. There are two key steps in our algorithm: dynamic clustering and selective perturbation, which we describe in detail as follows.
3.4.3.1 Dynamic Clustering

Considering that communities in social networks change significantly over time, we need to address the inconsistency problem by developing a dynamic community detection method. Dynamic clustering aims to find community structures in evolving graphs by simultaneously considering consecutive graphs in its clustering algorithms. There are several methods in the literature to cluster evolving graphs [7], but we found them to be unsuitable for use in our perturbation mechanism. One approach to dynamic clustering involves performing community detection at each timestamp independently, and then establishing relationships between communities to track their evolution [7]. We found that this approach suffers from performance issues induced by inherent randomness in clustering algorithms, in addition to the increased computational complexity.

Another approach is to combine multiple graphs into a single coupled graph [7]. The coupled graph is constructed by adding edges between the same nodes across different graphs. Clustering can be performed on the single coupled graph. We found that the clustering performance is very sensitive to the weights of the added links, resulting in unstable clustering results. Furthermore, the large dimensionality of the coupled graph significantly increases the computational overhead.

For our perturbation mechanism, we develop an adaptive dynamic clustering approach for clustering the graph \( G_t \) using the clustering result for the previous graph \( G_{t-1} \). This enables our perturbation mechanism to (a) exploit the link correlation/similarity in consecutive graph snapshots, and (b) reduce computation complexity by avoiding repeated clustering for unchanged links.

Clustering the graph \( G_t \) from the clustering result of the previous graph \( G_{t-1} \) requires a backtracking strategy. We use the maximum-modularity method [100] for clustering, which is hierarchical and thus easy to backtrack. Our backtrack strategy
is to first maintain a history of the merge operations that led to the current clustering. When an evolution occurs, the algorithm backtracks over the history of merge operations, in order to incorporate the new additions and deletions in the graph.

More concretely, if the link between node $x$ and node $y$ is changed (added or deleted), we omit all the $m$-hop neighborhoods of $x$ and $y$ as well as $x$ and $y$ themselves from the clustering result of the previous timestamp, and then perform re-clustering. All the new nodes, the changed nodes and their $m$-hop neighbors, and the remaining merged nodes in the previous clustering result would be considered as basic elements for clustering $G_t$ (recall Figure 3.2).

For efficient implementation, we store the intermediate results of the hierarchical clustering process in a data structure. Upon link changes between $x, y$, we free the $m$-hop neighborhood of $x, y$ from the stored data structure.

### 3.4.3.2 Selective Perturbation

**Intra-cluster Perturbation:** After clustering $G_t$ based on $G_{t-1}$ using our dynamic clustering method, we perturb $G_t$ based on $G_{t-1}$ and the perturbed $G'_{t-1}$. First, we compare the communities detected in $G_{t-1}$ and $G_t$, and classify them as changed or unchanged. Our unchanged classification does not require that the communities are exactly the same, but that the overlap among vertices/links exceeds a threshold. Our key idea is to keep the perturbation process for links in the unchanged communities to be identical to their perturbation in the previous snapshot. In this manner, we can preserve the privacy of these unchanged links to the largest extent; it is easy to see that alternate approaches would leak more information. For the communities which are classified as changed, our approach is to perturb their links independently of the perturbation in the previous timestamp. For independent perturbations,
we leverage the static perturbation method of Mittal et al. in [95]. Their static perturbation deletes all the edges in the original graph, and replaces each edge \((v, u)\) with a fake edge \((v, w)\) selected from the \(k\)-hop random walk starting from \(v\). Larger perturbation parameter \(k\) corresponds to better privacy and leads to worse utility.

**Inter-cluster Perturbation:** Finally, we need to interconnect the subgraphs identified above. Suppose that \(|v_a|\) nodes and \(|v_b|\) nodes are connecting communities \(a\) and \(b\) respectively, and they construct an inter-community subgraph. For each marginal node \(v_a(i) \in v_a\) and \(v_b(j) \in v_b\) (here the marginal node in community \(a\) (resp.\(b\)) refers to the node that has neighbors in the other community \(b\) (resp.\(a\)) ), we randomly connect them with probability \(\frac{\deg(v_a(i)) \deg(v_b(j)) |v_a|}{|E_{ab}|(|v_a|+|v_b|)}\). This probability is set for the preservation of degree distributions as analyzed in Section 3.7. Here, all the computations for \(\deg(\cdot), |v(\cdot)|, |E|\) only consider the marginal nodes. We can combine the perturbed links corresponding to the unchanged communities, changed communities, and inter-community subgraphs, to compute the output of our algorithm, i.e., \(G'_t\).

LinkMirage not only preserves the structural characteristics of the original graph series, but also protects the privacy of the users by randomizing the original links. As compared to prior work, our method provides stronger privacy and utility guarantees for evolving graphs. Detailed procedures are stated in Algorithm. 1.

LinkMirage improves the state-of-the-art methods such as [95] by incorporating the temporal graph topology. Surprisingly, our approach of first isolating communities and then selectively perturbing them provides benefits even in a static context! This is because previous static approaches use a single parameter to control the privacy/utility trade-off. Thus, if we apply them to the whole graph using high privacy parameters, it would destroy graph utility (e.g. community structures). On the other hand, LinkMirage applies perturbations selectively to communities; thus it is possible
to use a very high privacy parameter in the perturbation process, while preserving structural properties such as community structures.

**Algorithm 1** LinkMirage, with dynamic clustering (steps 1-2) and selective perturbation (steps 3-6). The parameter $k$ denotes the perturbation level for each community. Here, $ch$, $un$, $in$ are short for changed, unchanged, inter-community, respectively.

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>${G_t, G_{t-1}, G'_{t-1}}$ if $t \geq 1$ or ${G_t}$ if $t = 0$;</td>
<td>$G'_t; G'_t, C_t = \text{null}$;</td>
</tr>
<tr>
<td>if $t=0$;</td>
<td>cluster $G_0$ to get $C_0$;</td>
</tr>
<tr>
<td></td>
<td>label $C_0$ as changed, i.e. $C_0_ch = C_0$;</td>
</tr>
<tr>
<td>endif</td>
<td>/* Begin Dynamic Clustering */</td>
</tr>
<tr>
<td>/* Begin Dynamic Clustering */</td>
<td>1. free the nodes within $m$ hops of the changed links;</td>
</tr>
<tr>
<td></td>
<td>2. re-cluster the new nodes, the freed nodes, the remaining merged virtual nodes in $C_{(t-1)}$ to get $C_t$;</td>
</tr>
<tr>
<td>/* End Dynamic Clustering */</td>
<td>/* Begin Selective Perturbation */</td>
</tr>
<tr>
<td>/* Begin Selective Perturbation */</td>
<td>3. find the unchanged communities $C_{t-un}$ and the changed communities $C_{t-ch}$;</td>
</tr>
<tr>
<td></td>
<td>4. let $G'<em>{t-un} = G'</em>{(t-1)-un}$;</td>
</tr>
<tr>
<td></td>
<td>5. perturb $C_{t-ch}$ for $G'_{t-ch}$ by the static method;</td>
</tr>
<tr>
<td></td>
<td>6. foreach community pair $a$ and $b$:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>/* End Selective Perturbation */</td>
</tr>
</tbody>
</table>

3.4.4 Scalable Implementation

Our algorithm relies on two key graph theoretical techniques: community detection (serves as a foundation for the dynamic clustering step in LinkMirage) and ran-
dom walk (serves as a foundation for the selective perturbation step in LinkMirage). The computational complexity for both community detection and random walk is $O(|E_t|)$ \cite{7,95} where $|E_t|$ is the number of edges in graph $G_t$, therefore the overall computational complexity of our approach is $O(|E_t|)$. Furthermore, our algorithms are parallelizable. We adopt the GraphChi parallel framework in \cite{75} to implement our algorithm efficiently using a commodity workstation (3.6 GHz, 24GB RAM). Our parallel implementation scales to very large social networks; for example, the running time of LinkMirage is less than 100 seconds for the large scale Google+ dataset (940 million links) (will be described in Section 3.5.1) using our commodity workstation.

Table 3.2. Temporal Statistics of the Facebook Dataset.

<table>
<thead>
<tr>
<th>Time</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td># of nodes</td>
<td>9,586</td>
<td>9,719</td>
<td>11,649</td>
<td>13,848</td>
<td>14,210</td>
<td>16,344</td>
<td>18,974</td>
<td>26,220</td>
<td>35,048</td>
</tr>
<tr>
<td># of edges</td>
<td>48,966</td>
<td>38,058</td>
<td>47,024</td>
<td>54,787</td>
<td>49,744</td>
<td>58,099</td>
<td>65,604</td>
<td>97,095</td>
<td>142,274</td>
</tr>
<tr>
<td>Average degree</td>
<td>5.11</td>
<td>3.91</td>
<td>4.03</td>
<td>3.96</td>
<td>3.50</td>
<td>3.55</td>
<td>3.46</td>
<td>3.70</td>
<td>4.06</td>
</tr>
</tbody>
</table>

3.4.5 Visual Depiction

For our experiments, we consider a real world Facebook social network dataset \cite{124} among New Orleans regional network, spanning from September 2006 to January 2009. Here, we utilize the wall post interaction data which represents stronger trust relationships and comprises of 46,952 nodes (users) connected by 876,993 edges. We partitioned the dataset using three month intervals to construct a total of 9 graph instances as shown in Table 3.2. Fig. 3.3 depicts the outcome of our perturbation algorithm on the partitioned Facebook graph sequence with timestamp $t = 3, 4, 5$ (out of 9 snapshots), for varying perturbation parameter $k$ (perturbation parameter for each community). For comparative analysis, we consider a baseline approach \cite{95} that applies static perturbation for each timestamp independently. In the dynamic
clustering step of our experiments, we free the two-hop neighborhoods of the changed
nodes, i.e. \( m = 2 \).

![Figure 3.3. Visualization of LinkMirage on Facebook Dataset for \( t = 3, 4, 5 \).](image)

On the left, we can see that LinkMirage has superior utility than the baseline approach (Mittal et al.), especially for larger values of \( k \) (due to dynamic clustering). On the right, we show the overlapped edges (black) and the changed edges (yellow) between consecutive graphs: \( t=(3, 4) \) and \( t=(4, 5) \). We can see that in LinkMirage, the perturbation of unchanged communities is correlated across time (selective perturbation), minimizing information leakage and enhancing privacy.

The maximum-modularity clustering method yields two communities for \( G_3 \), three communities for \( G_4 \), and four communities for \( G_5 \). For the perturbed graphs, we use the same color for the vertices as in the original graph and we can see that fine-grained structures (related to utility) are preserved for both algorithms under small perturbation parameter \( k \), even though links are randomized. Even for high values of \( k \), LinkMirage can preserve the macro-level (such as community-level) structural characteristics of the graph. On the other hand, for high values of \( k \), the static perturbation algorithm results in the loss of structure properties, and appears to resemble a random graph. Thus, our approach of first isolating communities and applying perturbation at the level of communities has benefits even in a static context.

Fig. 3.3 also shows the privacy benefits of our perturbation algorithm for timestamps \( t = 4, 5 \). We can see that LinkMirage reuses perturbed links (shown as black unchanged links) in the unchanged communities (one unchanged community for \( t = 4 \) and two unchanged communities for \( t = 5 \)). Therefore, LinkMirage preserves the privacy of users’ social relationships by considering correlations among the graph se-
quence, and this benefit does not come at the cost of utility. In the following sections, we will formally quantify the privacy and utility properties of LinkMirage.

3.4.6 Supporting Applications

As discussed in Section 3.1, LinkMirage supports three types of applications: 1) Global access to obfuscated graphs: real-world applications can utilize our protocol to automatically obtain the secure social graphs to enable social relationships based systems. For instance, Tor operators [32] (or other anonymous communication network such as Pisces in [96]) can leverage the perturbed social relationships to set up the anonymous circuit; 2) Local access to the obfuscated graphs: an individual user can query our protocol for his/her perturbed friends (local neighborhood information), to implement distributed applications such as SybilLimit in [135]; 3) Mediated data analysis: the OSN providers can also publish perturbed graphs by leveraging LinkMirage to facilitate privacy-preserving data-mining research, i.e., to implement graph analytics such as pagerank score [104], modularity [100], while mitigating disclosure of users’ social relationships. Existing work in [34, 38] demonstrated that the implementation of graph analytic algorithms would leak certain information. To avoid repeatedly adding perturbations to the output of every graph analytic algorithm, which is rather costly, the OSN providers can first obtain the perturbed graphs by leveraging LinkMirage and then enable these graph analytics in a privacy-preserving manner.
3.5 Privacy Analysis

We now address the question of understanding link privacy of LinkMirage. We propose three privacy metrics: anti-inference privacy, indistinguishability, anti-aggregation privacy to evaluate the link privacy provided by LinkMirage. Both theoretical analysis and experimental results with a Facebook dataset (870K links) and a large-scale Google+ dataset (940M links) show the benefits of LinkMirage over previous approaches. We also illustrate the relationship between our privacy metric and differential privacy.

Table 3.3. Temporal Statistics of the Google+ Dataset.

<table>
<thead>
<tr>
<th>Time</th>
<th>Jul.29</th>
<th>Aug.8</th>
<th>Aug.18</th>
<th>Aug.28</th>
</tr>
</thead>
<tbody>
<tr>
<td># of nodes</td>
<td>16,165,781</td>
<td>17,483,936</td>
<td>17,850,948</td>
<td>19,406,327</td>
</tr>
<tr>
<td># of edges</td>
<td>505,527,124</td>
<td>560,576,194</td>
<td>575,345,552</td>
<td>654,523,658</td>
</tr>
<tr>
<td>Average degree</td>
<td>31.2714</td>
<td>32.0624</td>
<td>32.2305</td>
<td>33.7273</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time</th>
<th>Sep.7</th>
<th>Sep.17</th>
<th>Sep.27</th>
<th>Oct.7</th>
</tr>
</thead>
<tbody>
<tr>
<td># of nodes</td>
<td>19,954,197</td>
<td>24,235,387</td>
<td>28,035,472</td>
<td>28,942,911</td>
</tr>
<tr>
<td># of edges</td>
<td>686,709,660</td>
<td>759,226,300</td>
<td>886,082,314</td>
<td>947,776,172</td>
</tr>
<tr>
<td>Average degree</td>
<td>34.4143</td>
<td>31.3272</td>
<td>31.6058</td>
<td>32.7464</td>
</tr>
</tbody>
</table>

3.5.1 Experimental Datasets

To illustrate how the temporal information degrades privacy, we consider two social network datasets. The first one is a large-scale Google+ dataset whose temporal statistics are illustrated in Table 3.3. To the best of our knowledge, this is the largest temporal dataset of social networks in public domain. The Google+ dataset is crawled from July 2011 to October 2011 which has 28,942,911 nodes and 947,776,172 edges. The dataset only considers link additions, i.e. all the edges in the previous graphs exist in the current graph. We partitioned the dataset into 84 timestamps.

The second one is the 9-timestamp Facebook wall posts dataset as we stated in Section 3.4.5 with temporal characteristics shown in Table 3.2. It is worth noting
that the wall-posts data experiences tremendous churn with only 45% overlap for consecutive graphs. Since our dynamic perturbation method relies on the correlation between consecutive graphs, the evaluation of our dynamic method on the Facebook wall posts data is conservative. To show the improvement in performance of our algorithm for graphs that evolve at a slower rate, in actual applications, the graph sequence is likely to evolve in a much slower rate, which would show better performance by our dynamic perturbation method. For our experiments, we also consider a sampled graph sequence extracted from the Facebook wall posts data with 80% overlap for consecutive graphs.

3.5.2 Anti-Inference Privacy

First, we consider adversaries that aim to infer link information by leveraging Bayesian inference. We define the privacy of a link \( L_t \) (or a subgraph) in the \( t \)-th graph instance, as the difference between the posterior probability and the prior probability of the existence of the link (or a subgraph), computed by the adversary using its prior information \( W \), and the knowledge of the perturbed graph sequence \( \{G'_i\}_{i=0}^t \). Utilizing Bayesian inference, we have

**Definition 4.** For link \( L_t \) in the original graph sequence \( G_0, \ldots, G_t \) and the adversary’s prior information \( W \), the anti-inference privacy \( \text{Privacy}_{ai} \) for the perturbed graph sequence \( G'_0, \ldots, G'_t \) is evaluated by the similarity between the posterior probability \( P(L_t|\{G'_i\}_{i=0}^t, W) \) and the prior probability \( P(L_t|W) \), where the posterior probability is

\[
P(L_t|\{G'_i\}_{i=0}^t, W) = \frac{P(\{G'_i\}_{i=0}^t|L_t, W) \times P(L_t|W)}{P(\{G'_i\}_{i=0}^t|W)}
\]

Higher similarity implies better anti-inference privacy.
The difference between the posterior probability and the prior probability represents the information leaked by the perturbation mechanism. Similar intuition has been mentioned in [81]. Therefore, the posterior probability should not differ much from the prior probability.

In the above expression, $P(L_t|W)$ is the prior probability of the link, which can be computed based on the known structural properties of social networks, for example, by using link prediction algorithms [82]. Note that $P(\{G_i^\prime\}_i^t|W)$ is a normalization constant that can be analyzed by sampling techniques. The key challenge is to compute $P(\{G_i^\prime\}_i^t|L_t, W)$. The detailed process for computing the posterior probability can be found in [95].

For evaluation, we consider a special case where the adversary’s prior is the entire time series of original graphs except the link $L_t$ (which is the link we want to quantify privacy for, and $L_t = 1$ denotes the existence of this link while $L_t = 0$ denotes the non-existence of this link). Such prior information can be extracted from personal public information, Facebook related information or other application-related information as stated in [14]. Note that this is a very strong adversarial prior, which would lead to the worst-case analysis of link privacy. Denoting $\{\tilde{G}_i(L_t)\}_i^t$ as the prior which contains all the information except $L_t$, we have the posterior probability of link $L_t$ under the worst case is

$$
P(L_t|\{G_i^\prime\}_i^t, \{\tilde{G}_i(L_t)\}_i^t) = \frac{P(\{G_i^\prime\}_i^t, L_t, \{\tilde{G}_i(L_t)\}_i^t) \times P(L_t|\{\tilde{G}_i(L_t)\}_i^t)}{P(\{G_i^\prime\}_i^t|\{\tilde{G}_i(L_t)\}_i^t)}$$

where

$$P(\{G_i^\prime\}_i^t|L_t, \{\tilde{G}_i(L_t)\}_i^t) = P(G_0^\prime|\tilde{G}_0(L_t)) \times$$

$$P(G_1^\prime|G_0^\prime, \tilde{G}_0(L_t), \tilde{G}_1(L_t)) \cdots P(G_t^\prime|G_{t-1}^\prime, \tilde{G}_{t-1}(L_t), \tilde{G}_t(L_t))$$
Therefore, the objective of perturbation algorithms is to make $P(L_t | \{G'_i\}_{i=0}^t, \{\tilde{G}_i(L_t)\}_{i=0}^t)$ close to $P(L_t | \{\tilde{G}_i(L_t)\}_{i=0}^t)$.

**Comparison with previous work:** Fig. 3.4 shows the posterior probability distribution for the whole Facebook graph sequence and the sampled Facebook graph sequence with 80% overlapping ratio, respectively. We computed the prior probability using the link prediction method in [82]. We can see that the posterior probability corresponding to LinkMirage is closer to the prior probability than that of the method of Mittal et al. [95]. In Fig. 3.4(b), taking the point where the link probability equals 0.1, the distance between the posterior CDF and the prior CDF for the static approach is a factor of 3 larger than LinkMirage ($k = 20$). Larger perturbation degree $k$ improves privacy and leads to smaller difference with the prior probability. Finally, by comparing Fig. 3.4(a) and (b), we can see that larger overlap in the graph sequence improves the privacy benefits of LinkMirage.

![Figure 3.4](image_url)

Figure 3.4. (a),(b) represent the link probability distributions for the whole Facebook interaction dataset and the sampled Facebook interaction dataset with 80% overlap. We can see that the posterior probability of LinkMirage is more similar to the prior probability than the baseline approach.
We also compare with the work of Hay et al. in [56], which randomizes the graph with $r$ real links deleted and another $r$ fake links introduced. The probability for a real link to be preserved in the perturbed graph is $1 - r/m$, which should not be small otherwise the utility would not be preserved. Even considering $r/m = 0.5$ (which would substantially hurt utility [56]), the posterior probability for a link using the method of Hay et al. would be 0.5, *even without prior information*. In contrast, our analysis for LinkMirage considers a worst-case prior, and shows that the posterior probability is smaller than 0.5 for more than 50% of the links when $k = 20$ in Fig. 3.4.

Therefore, our LinkMirage provides significantly higher privacy than the work of Hay et al.

![Figure 3.5](image_url)

**Figure 3.5.** *Link probability distribution for the Google+ dataset under the adversary’s prior information extracted from the social-attribute network model in [50].*

**Adversaries with structural and contextual information:** Note that our analysis so far focuses on quantifying link-privacy under an adversary with prior information about the original network structure (including link prediction capabilities). In addition, some adversaries may also have access to contextual information about users in the social network, such as user attributes, which can also be used to predict network links (e.g., social-attribute network prediction model in [50]). We further computed the prior probability using such social-attribute network prediction model in [50] and showed the link probability for the Google+ dataset in Fig. 3.5.
The posterior probability of our LinkMirage is closer to the prior probability and thus LinkMirage achieves better privacy performance than previous work.

3.5.3 Indistinguishability

Based on the worse-case posterior probability of a link \( P(L_t|\{G'_i\}_{i=0}^t, \{\tilde{G}_i(L_t)\}_{i=0}^t) \), we need to qualify the privacy metric for adversaries who aim to distinguish the posterior probability with the prior probability. Since our goal is to reduce the information leakage of \( L_t \) based on the perturbed graphs \( \{G'_i\}_{i=0}^t \) and the prior knowledge \( \{\tilde{G}_i(L_t)\}_{i=0}^t \), we consider the metric of indistinguishability to quantify privacy, which can be evaluated by the conditional entropy of a private message given the observed variables \(^{[25]}\). The objective for an obfuscation scheme is to maximize the indistinguishability of the unknown input \( I \) given the observables \( O \), i.e. \( H(I|O) \) (where \( H \) denotes entropy of a variable \(^{[25]}\)). Here, we define our metric for link privacy as

**Definition 5.** The indistinguishability for a link \( L_t \) in the original graph \( G_t \) that the adversary can infer from the perturbed graph \( G'_t \) under the adversary’s prior information \( \{\tilde{G}_i(L_t)\}_{i=0}^t \) is defined as

\[
\text{Privacy}_{id} = H(L_t|\{G'_i\}_{i=0}^t, \{\tilde{G}_i(L_t)\}_{i=0}^t).
\]

Furthermore, we quantify the behavior of indistinguishability over time. For our analysis, we continue to consider the worst case prior of the adversary knowing the entire graph sequence except the link \( L_t \). To make the analysis tractable, we add another condition that if the link \( L \) exists, then it exists in all the graphs (link deletions are rare in real world social networks). For a large-scale graph, only one link would not affect the clustering result. Then, we have

**Theorem 2.** The indistinguishability decreases with time,

\[
H(L|\{G'_i\}_{i=0}^t, \{\tilde{G}_i(L)\}_{i=0}^t) \geq H(L|\{G'_i\}_{i=0}^{t+1}, \{\tilde{G}_i(L)\}_{i=0}^{t+1}) \quad (3.2)
\]
The inequality follows from the theorem *conditioning reduces entropy* in [25]. Eq.3.2 shows that the *indistinguishability* would not increase as time evolves. The reason is that over time, multiple perturbed graphs can be used by the adversary to infer more information about link \( L \).

Next, we theoretically show why LinkMirage has better privacy performance than the static method. For each graph \( G_t \), denote the perturbed graphs using LinkMirage and the static method as \( G'_t, G''_t \), respectively.

**Theorem 3.** The *indistinguishability* for LinkMirage is greater than that for the static perturbation method, i.e.

\[
H(L_t|\{G'_i\}_{i=0}^t, \{\tilde{G}_i(L_t)\}_{i=0}^t) \geq H(L_t|\{G''_i\}_{i=0}^t, \{\tilde{G}_i(L_t)\}_{i=0}^t) \tag{3.3}
\]

**Proof.** In LinkMirage, the perturbation for the current graph \( G_t \) is based on perturbation for \( G_{t-1} \). Let us denote the *changed* subgraph between \( G_{t-1}, G_t \) as \( G_{t-ch} \), then

\[
H(L_t|\{G'_i\}_{i=0}^t, \{\tilde{G}_i(L_t)\}_{i=0}^t)
=H(L_t|\{G'_i\}_{i=0}^{t-2}, G'_t, G_t - G_{t-ch}, G''_t, \{\tilde{G}_i(L_t)\}_{i=0}^t)
=H(L_t|\{G'_i\}_{i=0}^{t-1}, G''_t, \{\tilde{G}_i(L_t)\}_{i=0}^t)
\geq H(L_t|\{G''_i\}_{i=0}^{t-1}, G'_t, \{\tilde{G}_i(L_t)\}_{i=0}^t)
\geq H(L_t|\{G''_i\}_{i=0}^t, \{\tilde{G}_i(L_t)\}_{i=0}^t)
\]

where the first inequality also comes from the theorem *conditioning reduces entropy* in [25]. The second inequality generalizes the first inequality from a snapshot \( t \) to the entire sequence. From Eq.3.3 we can see that LinkMirage may offer superior *indistinguishability* compared to the static perturbation, and thus provides higher privacy. \( \square \)
(a) Timestamp $t$

Indistinguishability

$k=5$, Mittal et al.

$k=5$, LinkMirage

$k=20$, Mittal et al.

$k=20$, LinkMirage

Hay’s et al.

(b) Timestamp $t$

Indistinguishability

$k=5$, Mittal et al.

$k=5$, LinkMirage

$k=20$, Mittal et al.

$k=20$, LinkMirage

Hay et al.

Figure 3.6. (a),(b) represent the temporal indistinguishability for the whole Facebook interaction dataset and the sampled Facebook interaction dataset with 80% overlap. Over time, the adversary has more information, resulting in decreased indistinguishability. We can also see that LinkMirage has higher indistinguishability than the static method and the Hay’s method in [56], although it still suffers from some information leakage. Comparing (a) and (b), we can see that the advantage of LinkMirage is more obvious for larger overlapped graph sequence.

Comparison with previous work: Next, we experimentally analyze our indistinguishability metric over time. Fig. 3.6 depicts the indistinguishability metric using the whole Facebook graph sequence and the sampled Facebook graph sequence with 80% overlap. We can see that the static perturbation leaks more information over time. In contrast, the selective perturbation achieves significantly higher indistinguishability. In Fig. 3.6(a), after 9 snapshots, and using $k = 5$, the indistinguishability of the static perturbation method is roughly $1/10$ of the indistinguishability of LinkMirage. This is because selective perturbation explicitly takes the temporal evolution into consideration, and stems privacy degradation via the selective perturbation step. Comparing Fig. 3.6(a) and (b), LinkMirage has more advantages for larger overlapped graph sequence.
We also compare with the work of Hay et al. in [56]. For the first timestamp, the probability for a real link to be preserved in the anonymized graph is $1 - r/m$. As time evolves, the probability would decrease to $(1 - r/m)^t$. Combined with the prior probability, the corresponding indistinguishability for the method of Hay et al. is shown as the black dotted line in Fig. 3.6, which converges to 0 very quickly (we also consider $r/m = 0.5$ which would substantially hurt utility [56]). Compared with the work of Hay et al, LinkMirage significantly improves privacy performance. Even when $t = 1$, LinkMirage with $k = 20$ achieves up to 10x improvement over the approach of Hay et al. in the indistinguishability performance.

### 3.5.4 Anti-aggregation Privacy

Next, we consider the adversaries who try to aggregate all the previously published graphs to infer more information. Recall that after community detection in our algorithm, we anonymize the links by leveraging the $k$-hop random walk. Therefore, the perturbed graph $G'$ is actually a sampling of the $k$-hop graph $G^k$, where the $k$-hop graph $G^k$ represents graph where all the $k$-hop neighbors in the original graph are connected. It is intuitive that a larger difference between $G^k$ and $G'$ represents better privacy. Here, we utilize the distance between the corresponding transition probability matrices $\|P^k_t - P'_t\|_{TV}$ to measure this difference. Specifically, we choose the total variance distance to evaluate the statistical distance between $P^k_t$ and $P'_t$ as in [95].

And we extend the definition of total variance [60] from vector to matrix by averaging total variance distance of each row in the matrix, i.e. $\|P^k_t - P'_t\|_{TV} = \frac{1}{|V_t|} \sum_{v=1}^{|V_t|} \|P^k_t(v) - P'_t(v)\|_{TV}$, where $P^k_t(v), P'_t(v)$ denotes the $v$-th row of $P^k_t, P'_t$. We then formally define the anti-aggregation privacy as
Definition 6. The anti-aggregation privacy for a perturbed graph $G'_t$ with respect to the original graph $G_t$ and the perturbation parameter $k$ is 
$$\text{Privacy}_{\text{aa}}(G_t, G'_t, k) = \|P^k_t - P'_t\|_{TV}.$$ 

The adversary’s final objective is to obtain an estimated measurement of the original graph, e.g. the estimated transition probability matrix $\hat{P}_t$ which satisfies $\hat{P}^k_t = P'_t$. A straightforward manner to evaluate privacy is to compute the estimation error of the transition probability matrix i.e. $\|P_t - \hat{P}_t\|_{TV}$. We can derive the relationship between the anti-aggregation privacy and the estimation error as

Theorem 4. The anti-aggregation privacy is a lower bound of the estimation error for the adversaries, and

$$\|P^k_t - P'_t\|_{TV} \leq k\|P_t - \hat{P}_t\|_{TV}$$  \hspace{1cm} (3.4)

Proof.

$$\|P^k_t - P'_t\|_{TV}$$
$$=\|P^k_t - \hat{P}^k_t\|_{TV}$$
$$\leq \frac{1}{2|V_t|} \sum_{v=1}^{|V_t|} \|P_t(v)P^{k-1}_t - \hat{P}_t(v)\hat{P}^{k-1}_t\|_1 + \frac{1}{2|V_t|} \sum_{v=1}^{|V_t|} \|\hat{P}_t(v)P^{k-1}_t - \hat{P}_t(v)\hat{P}^{k-1}_t\|_1$$
$$=\|P_t - \hat{P}_t\|_{TV} + \|P^{k-1}_t - \hat{P}^{k-1}_t\|_{TV}$$
$$\leq k\|P_t - \hat{P}_t\|_{TV}$$ \hspace{1cm} (3.5)

We further consider the network evolution where the adversary can combine all the perviously perturbed graphs together to extract more $k$-hop information of the current graph. Under this situation, a strategic methodology for the adversary is to combine the perturbed graph series $G'_0, \cdots, G'_t$, to construct a new perturbed graph $\tilde{G}'_t$, where
\[ \tilde{G}_t = \bigcup_{i=0,1,\ldots,t} G_i' \] The combined perturbed graph \( \tilde{G}_t' \) contains more information about the \( k \)-hop graph \( G^k_t \) than \( G_t' \). Correspondingly, the transition probability matrix \( \tilde{P}_t' \) of the combined perturbed graph \( \tilde{G}_t' \) would provide more information than \( P_t' \). That is to say, the \textit{anti-aggregation privacy} decreases with time.

\[
\begin{array}{c}
\text{Anti-aggregation Privacy} \\
\hline
0 & 0.2 & 0.4 & 0.6 & 0.8 & 1 \\
\hline
\text{(a) Timestamp t} \\
\end{array}
\]

\[
\begin{array}{c}
\text{Anti-aggregation Privacy} \\
\hline
0.6 & 0.65 & 0.7 & 0.75 & 0.8 & 0.85 \\
\hline
\text{(b) Timestamp t} \\
\end{array}
\]

Figure 3.7. (a)(b) show the temporal anti-aggregation privacy for the Google+ dataset and the Facebook dataset, respectively. The anti-aggregation privacy decreases as time evolves because more information is leaked with more perturbed graphs available. Leveraging selective perturbation, LinkMirage achieves much better anti-aggregation privacy than the static baseline method.

**Comparison with previous work:** We evaluate the anti-aggregation privacy of LinkMirage on both the Google+ dataset and the Facebook dataset. Here we perform our experiments based on a conservative assumption that a link always exists after it is introduced. The anti-aggregation privacy decreases with time since more information about the \( k \)-hop neighbors of the graph is leaked as shown in Fig. 3.7. Our selective perturbation preserves correlation between consecutive graphs, therefore leaks less information and achieves better privacy than the static baseline method. For the
Google+ dataset, the anti-aggregation privacy for the method of Mittal et al. is only 1/10 of LinkMirage after 84 timestamps.

### 3.5.5 Relationship with Differential Privacy

Our anti-inference privacy analysis considers the worst-case adversarial prior to infer the existence of a link in the graph. Next, we uncover a novel relationship between this anti-inference privacy and differential privacy.

Differential privacy is a popular theory to evaluate the privacy of a perturbation scheme [34, 38]. The framework of differential privacy defines local sensitivity of a query function \( f \) on a dataset \( D \) as the maximal \( |f(D_1) - f(D_2)|_1 \) for all \( D_2 \) differing from \( D_1 \) in at most one element \( df = \max_{D_2} \|f(D_1) - f(D_2)\|_1 \). Based on the theory of differential privacy, a mechanism that adds independent Laplacian noise with parameter \( df/\epsilon \) to the query function \( f \), satisfies \( \epsilon \)-differential privacy. The degree of added noise, which determines the utility of the mechanism, depends on the local sensitivity. To achieve a good utility as well as privacy, the local sensitivity \( df \) should be as small as possible. The following lemma demonstrates the effectiveness of our worst-case Bayesian analysis by showing that the objective for good utility-privacy balance under our worst-case Bayesian analysis is equivalent to that under differential privacy.

**Remark 1.** The requirement for good utility-privacy balance in differential privacy is equivalent to the objective of our Bayesian analysis under the worst case.

**Proof.** When considering differential privacy for a time series of graph sequence \( \{G_i\}_{i=0}^t \), we have
\[ f(D) = P(\{G_i^t\}_{i=0}^t | \{\tilde{G}_i(L_t)\}_{i=0}^t, L_t = 1) \]
\[ f(D') = P(\{G_i^t\}_{i=0}^t | \{\tilde{G}_i(L_t)\}_{i=0}^t, L_t = 0) \]  
(3.6)

For a good privacy performance, we need
\[ P(\{G_i^t\}_{i=0}^t | \{\tilde{G}_i(L_t)\}_{i=0}^t, L_t = 1) \approx P(\{G_i^t\}_{i=0}^t | \{\tilde{G}_i(L_t)\}_{i=0}^t, L_t = 0) \]  
(3.7)

Since the probability of \(\{G_i^t\}_{i=0}^t\) given \(\{\tilde{G}_i(L_t)\}_{i=0}^t\) as
\[ P(\{G_i^t\}_{i=0}^t | \{\tilde{G}_i(L_t)\}_{i=0}^t) = P(\{G_i^t\}_{i=0}^t | \{\tilde{G}_i(L_t)\}_{i=0}^t, L_t = 1) P(L_t = 1 | \{\tilde{G}_i(L_t)\}_{i=0}^t) 
+ P(\{G_i^t\}_{i=0}^t | \{\tilde{G}_i(L_t)\}_{i=0}^t, L_t = 0) P(L_t = 0 | \{\tilde{G}_i(L_t)\}_{i=0}^t) \]  
(3.8)

it is easy to see that if the condition for a good privacy performance holds, we have
\[ P(L_t | \{G_i^t\}_{i=0}^t, \{\tilde{G}_i(L_t)\}_{i=0}^t) = \frac{P(\{G_i^t\}_{i=0}^t | \{\tilde{G}_i(L_t)\}_{i=0}^t, L_t) \times P(L_t | \{\tilde{G}_i(L_t)\}_{i=0}^t)}{P(\{G_i^t\}_{i=0}^t | \{G_i(L_t)\}_{i=0}^t)} \approx P(L_t | \{\tilde{G}_i(L_t)\}_{i=0}^t) \]  
(3.9)

which is the same as in Definition 4 and means that the posterior probability is similar to the prior probability, i.e., the adversary is bounded in the information it can learn from the perturbed graphs.

3.5.6 Summary for Privacy Analysis

In summary, we have the following important observations for the privacy performance of LinkMirage:

\[ \square \]
• LinkMirage provides rigorous privacy guarantees to defend against adversaries who have prior information about the original graphs, and the adversaries who aim to combine multiple released graphs to infer more information.

• LinkMirage shows significant privacy advantages in anti-inference privacy, indistinguishability and anti-aggregation privacy, by outperforming previous methods by a factor up to 10.

3.6 Applications

Applications such as anonymous communication \cite{32, 96, 97} and vertex anonymity mechanisms \cite{86, 106, 138} can utilize LinkMirage to obtain the entire obfuscated social graphs. Alternatively, each individual user can query LinkMirage for his/her perturbed neighborhoods to set up distributed social relationship based applications such as SybilLimit \cite{135}. Further, the OSN providers can also leverage LinkMirage to perturb the original social topologies only once and support multiple privacy-preserving graph analytics, e.g., privately compute the pagerank/modularity of social networks.

Figure 3.8. The worst case probability of deanonymizing users’ communications ($f = 0.1$). Over time, LinkMirage provides better anonymity compared to the static approaches.
3.6.1 Anonymous Communication \[32, 96, 97\]

As a concrete application, we consider the problem of anonymous communication \[32, 96, 97\]. Systems for anonymous communication aim to improve user’s privacy by hiding the communication link between the user and the remote destination. Nagaraja et al. and others \[32, 96, 97\] have suggested that the security of anonymity systems can be improved by leveraging users’ trusted social contacts.

We envision that our work can be a key enabler for the design of such social network based systems, while preserving the privacy of users’ social relationships. We restrict our analysis to low-latency anonymity systems that leverage social links, such as the Pisces protocol \[96\].

Similar to the Tor protocol, users in Pisces rely on proxy servers and onion routing for anonymous communication. However, the relays involved in the onion routing path are chosen by performing a random walk on a trusted social network topology. Recall that LinkMirage better preserves the evolution of temporal graphs in Fig. 3.3. We now show that this translates into improved anonymity over time, by performing an analysis of the degradation of user anonymity over multiple graph snapshots. For each graph snapshot, we consider a worst case anonymity analysis as follows: if a user’s neighbor in the social topology is malicious, then over multiple communication rounds (within that graph instance) its anonymity will be compromised using state-of-the-art traffic analysis attacks \[130\]. Now, suppose that all of a user’s neighbors in the first graph instance are honest. As the perturbed graph sequence evolves, there is further potential for degradation of user anonymity since in the subsequent instances, there is a chance of the user connecting to a malicious neighbor. Suppose the probability for a node to be malicious is $f$. Denote $n_t(v)$ as the distinct neighbors of node $v$ at time $t$. For a temporal graph sequence, the number of the union neighbors $\bigcup_{k=0}^{t} n_k(v)$ of $v$ increases with time, and the probability for $v$ to be attacked under the
worst case is \( P_{attack}(v) = 1 - (1 - f)^{\sum_{k=0}^{n_k(v)}} \). Note that in practice, the adversary’s prior information will be significantly less than the worst-case adversary.

Fig. 3.8 depicts the degradation of the worst-case anonymity with respect to the number of perturbed topologies. We can see that the attack probability for our method is lower than the static approach with a factor up to 2. This is because over consecutive graph instances, the users’ social neighborhood has higher similarity as compared to the static approach, reducing potential for anonymity degradation. Therefore, LinkMirage can provide better security for anonymous communication, and other social trust based applications.

### 3.6.2 Vertex Anonymity [86, 106, 138]

Previous work for vertex anonymity [86, 106, 138] would be defeated by de-anonymization techniques [64, 98, 103, 115]. LinkMirage can serve as a formal first step for vertex anonymity, and even improve its defending capability against de-anonymization attacks. We apply LinkMirage to anonymize vertices, i.e., to publish a perturbed topology without labeling any vertex. In [64], Ji et al. modeled the anonymization as a sampling process where the sampling probability \( p \) denotes the probability of an edge in the original graph \( G_o \) to exist in the anonymized graph \( G' \). LinkMirage can also be applied for such model, where the perturbed graph \( G' \) is sampled from the \( k \)-hop graph \( G^k \) (corresponding to \( G_o \)).

They also derived a theoretical bound of the sampling probability \( p \) for perfect de-anonymization, and found that a weaker bound is needed with a larger value of the sampling probability \( p \). Larger \( p \) implies that \( G' \) is topologically more similar to \( G \), making it easier to enable a perfect de-anonymization. When considering social network evolution, the sampling probability \( p \) can be estimated as \( |E(G'_0, \ldots, G'_t)|/|E(G^k_0, \ldots, G^k_t)| \), where \( E(G'_0, \ldots, G'_t) \) are the edges of the
perturbed graph sequence, and \( E(G_0^k, \cdots, G_t^k) \) are the edges of the \( k \)-hop graph sequence. Compared with the static baseline approach, LinkMirage selectively reuses information from previously perturbed graphs, thus leading to smaller overall sampling probability \( p \), which makes it harder to perfectly de-anonymize the graph sequence. For example, the average sampling probability \( p \) for the Google+ dataset (with \( k = 2 \)) is 0.431 and 0.973 for LinkMirage and the static method respectively. For the Facebook temporal dataset (with \( k = 3 \)), the average sampling probability \( p \) is 0.00012 and 0.00181 for LinkMirage and the static method respectively. Therefore, LinkMirage is more resilient against de-anonymization attacks even when applied to vertex anonymity, with up to 10x improvement.

Figure 3.9. (a) shows the false positive rate for Sybil defenses. We can see that the perturbed graphs have lower false positive rate than the original graph. Random walk length is proportional to the number of Sybil identities that can be inserted in the system. (b) shows that the final attack edges are roughly the same for the perturbed graphs and the original graphs.
Next, we consider Sybil defenses systems which leverage the published social topologies to detect fake accounts in the social networks. Here, we analyze how the use of a perturbed graph changes the Sybil detection performance of SybilLimit, which is a representative Sybil defense system. Each user can query LinkMirage for his/her perturbed friends to set up the implementation of SybilLimit. Fig. 3.9(a) depicts the false positives (honest users misclassified as Sybils) with respect to the random walk length in the Sybillimit protocol. Fig. 3.9(b) shows the final attack edges with respect to the attack edges in the original topology. We can see that the false positive rate is much lower for the perturbed graphs than for the original graph, while the number of the attack edges stay roughly the same for the original graph and the perturbed graphs. The number of Sybil identities that an adversary can insert is given by \( S = g' \cdot w' \) (\( g' \) is the number of attack edges and \( w' \) is the random walk parameter in the protocol). Since \( g' \) stays almost invariant and the random walk parameter \( w' \) (for any desired false positive rate) is reduced, LinkMirage improves Sybil resilience and provides the privacy of the social relationships such that Sybil defense protocols continue to be applicable (similar to static approaches whose Sybil-resilience performance have been demonstrated in previous work).

<table>
<thead>
<tr>
<th></th>
<th>Original Graph</th>
<th>LinkMirage ( k = 2 )</th>
<th>LinkMirage ( k = 5 )</th>
<th>Mittal et al. ( k = 2 )</th>
<th>Mittal et al. ( k = 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google+</td>
<td>Modularity</td>
<td>0.605</td>
<td>0.601</td>
<td>0.603</td>
<td>0.591</td>
</tr>
<tr>
<td>Facebook</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modularity</td>
<td>Original Graph</td>
<td>0.488</td>
<td>0.479</td>
<td>0.487</td>
<td>0.476</td>
</tr>
</tbody>
</table>

Table 3.4. Modularity of Perturbed Graph Topologies
3.6.4 Privacy-preserving Graph Analytics \[100,104]\n
Next, we demonstrate that LinkMirage can also benefit the OSN providers for privacy-preserving graph analytics. Previous work in \[34,38\] have demonstrated that the implementation of graph analytic algorithms would also result in information leakage. To mitigate such privacy degradation, the OSN providers could add perturbations (noises) to the outputs of these graph analytics. However, if the OSN providers aim to implement multiple graph analytics, the process for adding perturbations to each output would be rather complicated. Instead, the OSN providers can first obtain the perturbed graph by leveraging LinkMirage and then set up these graph analytics in a privacy-preserving manner.

Here, we first consider the pagerank \[104\] as an effective graph metric. For the Facebook dataset, we have the average differences between the perturbed pagerank score and the original pagerank score as 0.0016 and 0.0018 for \(k = 5\) and \(k = 20\) respectively in LinkMirage. In comparison, the average differences are 0.0019 and 0.0087 for \(k = 5\) and \(k = 20\) in the approach of Mittal et al. LinkMirage preserves the pagerank score of the original graph with up to 4x improvement over previous methods. Next, we show the modularity \[100\] (computed by the timestamp \(t = 3\) in the Google+ dataset and the Facebook dataset, respectively) in Table 3.4. We can see that LinkMirage preserves both the pagerank score and the modularity of the original graph, while the method of Mittal et al. degrades such graph analytics especially for larger perturbation parameter \(k\) (recall the visual intuition of LinkMirage in Fig. 3.3).

3.6.5 Summary for Applications of LinkMirage

In summary, we have the following important observations for the applications of LinkMirage:
• LinkMirage preserves the privacy of users’ social contacts while enabling the design of social relationships based applications. Compared to previous methods, LinkMirage results in significantly lower attack probabilities (with a factor up to 2) when applied to anonymous communications and higher resilience to de-anonymization attacks (with a factor up to 10) when applied to vertex anonymity systems.

• LinkMirage even surprisingly improves the Sybil detection performance when applied to the distributed SybilLimit systems.

• LinkMirage preserves the utility performance for multiple graph analytics applications, such as pagerank score and modularity with up to 4x improvement.

3.7 Utility Analysis

Following the application analysis in Section 3.6, we aim to develop a general metric to characterize the utility of the perturbed graph topologies. Furthermore, we theoretically analyze the lower bound on utility for LinkMirage, uncover connections between our utility metric and structural properties of the graph sequence, and experimentally analyze our metric using the real-world Google+ and Facebook datasets.

3.7.1 Metrics

We aim to formally quantify the utility provided by LinkMirage to encompass a broader range of applications. One intuitive global utility metric is the degree of vertices. It is interesting to find that the expected degree of each node in the perturbed graph is the same as the original degree.
Theorem 5. The expected degree of each node after perturbation by LinkMirage is the same as in the original graph: \( \forall v \in V_t, E(\deg'(v)) = \deg(v) \), where \( \deg'(v) \) denotes the degree of vertex \( v \) in \( G'_t \).

Proof. According to Theorem 3 in [95], we have \( E(\deg'_\text{com}(v)) = \deg(v) \), where \( \deg'_\text{com}(v) \) denotes the degree of \( v \) after perturbation within community. Then we consider the random perturbation for inter-community subgraphs. Since the probability for an edge to be chosen is \( \frac{\deg(v_a(i)) \deg(v_b(j)) |v_a|}{|E_{ab}|(|v_a|+|v_b|)} \), the expected degree after inter-community perturbation satisfies

\[
E(\deg'_{\text{inter}}(v_a(i))) = \sum_j \frac{\deg(v_a(i)) \deg(v_b(j)) (|v_a|+|v_b|)}{|E_{ab}|(|v_a|+|v_b|)}
= \deg(v_a(i))
\]

Combining with the expectations under static scenario, we have \( E(\deg'(v)) = \deg(v) \).

To understand the utility in a fine-grained level, we further define our utility metric as

Definition 7. The Utility Distance (UD) of a perturbed graph sequence \( G'_0, \cdots, G'_T \) with respect to the original graph sequence \( G_0, \cdots, G_T \), and an application parameter \( l \) is defined as

\[
UD(G_0, \cdots G_T, G'_0, \cdots G'_T, l) = \frac{1}{T+1} \sum_{t=0}^{T} \sum_{v \in V_t} \frac{1}{|V_t|} \| P_t^l(v) - (P_t')^l(v) \|_{TV}
\]  

(3.11)

Our definition for utility distance in Eq. 3.11 is intuitively reasonable for a broad class of real-world applications, and captures the behavioral differences of \( l \)-hop random walks between the original graphs and the perturbed graphs. We note that
random walks are closely linked to the structural properties of social networks. In fact, a lot of social network based security applications such as Sybil defenses and anonymity systems directly perform random walks in their protocols. The parameter $l$ is application specific; for applications that require access to fine grained local structures, such as recommendation systems, the value of $l$ should be small. For other applications that utilize coarse and macro structure of the social graphs, such as Sybil defense mechanisms, $l$ can be set to a larger value (typically around 10 in). Therefore, this utility metric can quantify the utility performance of LinkMirage for various applications in a general manner.

Table 3.5. Graph Metrics of the Original and the Perturbed Graphs for the Google+ Dataset.

<table>
<thead>
<tr>
<th></th>
<th>Clustering Coefficient</th>
<th>Assortativity Coefficient</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Graph</td>
<td>0.2612</td>
<td>-0.0152</td>
</tr>
<tr>
<td>LinkMirage $k = 2$</td>
<td>0.2263</td>
<td>-0.0185</td>
</tr>
<tr>
<td>LinkMirage $k = 5$</td>
<td>0.1829</td>
<td>-0.0176</td>
</tr>
<tr>
<td>LinkMirage $k = 10$</td>
<td>0.0864</td>
<td>-0.0092</td>
</tr>
<tr>
<td>LinkMirage $k = 20$</td>
<td>0.0136</td>
<td>-0.0063</td>
</tr>
</tbody>
</table>

Note that LinkMirage is not limited to only preserving the community structure of the original graphs. We evaluate two representative graph theoretic metrics clustering coefficient and assortativity coefficient as listed in Table 3.5. We can see that LinkMirage well preserves such fine-grained structural properties for smaller perturbation parameter $k$. Therefore, the extent to which the utility properties are preserved depends on the perturbation parameter $k$.

### 3.7.2 Relationships with Other Graph Structural Properties

The mixing time $\tau(\epsilon) = \min_r \max_v (r \mid P_t^r(v) - \pi_t \mid_{TV} < \epsilon)$. Based on the Perron-Frobenius theory, we denote the eigenvalues of $P_t$ as $1 = \lambda_1 \geq$
\( \mu_1(G_t) \geq \mu_2(G_t) \geq \cdots \mu_{|V_t|}(G_t) \geq -1. \) The convergence rate of the Markov chain to \( \pi_t \) is determined by the second largest eigenvalue modulus (SLEM) as \( \mu(G_t) = \max\left(\mu_2(G_t), -\mu_{|V_t|}(G_t)\right) \).

Since our utility distance is defined by using the transition probability matrix \( P_t \), this metric can be proved to be closely related to structural properties of the graphs, as shown in Theorem 6 and Theorem 7. We defer the proofs to A.1 in the Appendix to improve readability.

**Theorem 6.** Let us denote the utility distance between the perturbed graph \( G'_t \) and the original graph \( G_t \) by \( UD(G_t, G'_t, l) \), then we have \( \tau_{G'_t}(UD(G_t, G'_t, \tau_{G_t}(\epsilon)) - \epsilon) \geq \tau_{G_t}(\epsilon) \).

**Theorem 7.** Let us denote the second largest eigenvalue modulus (SLEM) of transition probability matrix \( P_t \) of graph \( G_t \) as \( \mu_{G_t} \). We can bound the SLEM of a perturbed graph \( G'_t \) using the mixing time of the original graph, and the utility distance between the graphs as \( \mu_{G'_t} \geq 1 - \frac{\log n + \log UD(G_t, G'_t, \tau_{G_t}(\epsilon)) - \epsilon}{\tau_{G_t}(\epsilon)} \).

### 3.7.3 Upper Bound of Utility Distance

LinkMirage aims to limit the degradation of link privacy over time. Usually, mechanisms that preserve privacy trade-off application utility. In the following, we will theoretically derive an upper bound on the utility distance for our algorithm. This corresponds to a lower bound on utility that LinkMirage is guaranteed to provide.

**Theorem 8.** The utility distance of LinkMirage is upper bounded by \( 2l \) times the sum of the utility distance of each community \( \epsilon \) and the ratio cut \( \delta_t \) for each \( G_t \), i.e.

\[
 UD(G_0, \cdots G_T, G'_0, \cdots G'_T, l) \leq \frac{1}{T+1} \sum_{t=0}^{T} 2l(\epsilon + \delta_t)
\]  

(3.12)
where $\delta_t$ denotes the number of inter-community links over the number of vertices, and each community $C_{k(t)}$ within $G_t$ satisfies $\|C_{k(t)} - C'_{k(t)}\|_{TV} \leq \epsilon$. We defer the proofs to the A.2 in the Appendix to improve readability.

Note that an upper bound on utility distance corresponds to a lower bound on utility of our algorithm. While better privacy usually requires adding more noise to the original sequence to obtain the perturbed sequence, thus we can see that LinkMirage is guaranteed to provide a minimum level of utility performance.

In the derivation process, we do not take specific evolutionary pattern such as the overlapped ratio into consideration, therefore our theoretical upper bound is rather loose. Next, we will show that in practice, LinkMirage achieves smaller utility distance (higher utility) than the baseline approach of independent static perturbations.

![Utility Distance Graphs](image)

Figure 3.10. (a), (b) show the utility distances using the Google+ dataset and the Facebook dataset, respectively. Larger perturbation parameter $k$ results in larger utility distance. Larger application parameter $l$ decreases the distance, which shows the effectiveness of LinkMirage in preserving global community structures.
3.7.4 Utility Experiments Analysis

Fig. 3.10(a)(b) depict the utility distance for the Google+ and the Facebook graph sequences, for varying perturbation degree $k$ and the application level parameter $l$. We can also see that as $k$ increases, the distance metric increases. This is natural since additional noise increase the distance between probability distributions computed from the original and the perturbed graph series. As the application parameter $l$ increases, the distance metric decreases. This illustrates that LinkMirage is more suited for security applications that rely on macro structures, as opposed to applications that require exact information about one or two hop neighborhoods. Furthermore, our experimental results in Table 3.4 also demonstrate the utility advantage of our LinkMirage over the approach of Mittal et al. [95] in real world applications.

3.8 Chapter Summary

LinkMirage effectively mediates privacy-preserving access to users’ social relationships, since 1) LinkMirage preserves key structural properties in the social topology while anonymizing intra-community and inter-community links; 2) LinkMirage provides rigorous guarantees for the anti-inference privacy, indistinguishability and anti-aggregation privacy, in order to defend against sophisticated threat models for both static and temporal graph topologies; 3) LinkMirage significantly outperforms baseline static techniques in terms of both link privacy and utility, which have been verified both theoretically and experimentally using real-world Facebook dataset (with 870K links) and the large-scale Google+ dataset (with 940M links). LinkMirage enables the deployment of real-world social relationship based applications such as graph analytic, anonymity systems, and Sybil defenses while preserving the privacy of users’ social relationships.
Chapter 4

Dependence Makes You Vulnerable: Differential Privacy Under Dependent Tuples

Differential privacy (DP) is a widely accepted mathematical framework for protecting data privacy. Simply stated, it guarantees that the distribution of query results changes only slightly due to the modification of any one tuple in the database. This allows protection, even against powerful adversaries, who know the entire database except one tuple. For providing this guarantee, differential privacy metric/mechanism does not explicitly model dependence (correlation) existing in the database, which may lead to degradation in expected privacy levels especially when applied to real-world datasets that manifest natural dependence owing to various social, behavioral, and genetic relationships between users. In this work, we make several contributions that not only demonstrate the feasibility of exploiting the above vulnerability but also provide steps towards mitigating it. First, we present an inference attack, using real datasets, where an adversary leverages the probabilistic dependence between tu-
ples to extract users’ sensitive information from noisy query results generated from existing DP mechanisms. Second, we introduce the notion of dependent differential privacy (DDP) that accounts for the dependence/correlation that exists between tuples and propose a dependent perturbation mechanism (DPM) to achieve the privacy guarantees in DDP. Finally, using a combination of theoretical analysis and extensive experiments involving different classes of queries (e.g., machine learning queries, graph queries) issued over multiple large-scale real-world datasets, we show that our DPM consistently outperforms the state-of-the-art approaches in managing the privacy-utility tradeoffs for dependent/correlated data.

4.1 Motivation

The notion of differential privacy (DP) [18, 20, 34, 36, 39, 43], which provides a rigorous mathematical foundation for defining and preserving privacy, has received considerable attention among existing frameworks [15, 34, 80, 89, 95, 116]. Used for protecting the privacy of aggregate query results over statistical databases, DP guarantees that the distribution of query outputs changes only slightly with the modification of a single tuple in the database. Thus, the information that an adversary can infer through observing the query output is strictly bounded by a function of the privacy budget.

Existing DP metric/mechanism does not explicitly model tuple dependence (correlation) that occurs naturally in datasets due to social, behavioral and genetic interactions between users. For example, in a social network graph (with nodes representing users, and edges representing ‘friendship’ relations), the participation of one user could harm the privacy of other individuals based on the ‘friendship’ between them [82]. Private attributes in a user’s record can be inferred by exploiting the public attributes

\footnote{The material in this chapter is based on [83].}
of other users sharing similar interests [17]. A user’s susceptibility to a contagious
disease can be easily inferred by an adversary who has access to noisy query results
and is aware of the fact that the user’s immediate family members are part of the
database being queried [71]. Social and behavioral dependence have also been used
to perform de-anonymization attacks on released datasets [62, 64, 78, 98, 103, 115].

The fact that dependence (or correlation) among tuples can degrade the expected
privacy guarantees of existing DP mechanisms was first observed by Kifer et al. [71],
and later in [22, 57, 72, 139]. Based on our own experiments with real-world datasets
in Section 4.3, we attribute this degradation to a faster exhaustion of the privacy bud-
get in DP. In prior work, the Pufferfish framework [72], proposed as a generalization
of DP, incorporated adversarial belief about existing data relationships using a data
generation model maintained as a distribution over all possible database instances.
However, the framework did not propose any specific perturbation algorithm to han-
dle the dependence. The Blowfish framework [57], which is a subclass of the Pufferfish
framework, allowed users to specify adversarial knowledge about the database in the
form of deterministic policy constraints and provided perturbation mechanisms to
handle these constraints. Dependence between records have also been modeled in
the mathematical framework for membership privacy in [81]. However, again no spe-
cific algorithms have been proposed for achieving the privacy objectives. Finally, to
handle correlation in network data using DP, the authors in [22] multiplied the sens-
sitivity of the query output with the number of correlated records. This technique
resulted in excessive noise being added to the output severely degrading the utility
of the shared data. Zhu et al. [139] exploited the linear correlations among tuples
to evaluate their dependent relationships, which does not rigorously meet any pri-

vacy requirement and their perturbation algorithm for dependent tuples still need
further investigations. In this work, we formalize the notion of dependent differ-
ential privacy (DDP) to handle probabilistic dependence constraints between tuples

75
while providing rigorous privacy guarantees. We further develop an effective dependent perturbation mechanism (DPM) to achieve the privacy guarantees in DDP. Our mechanism uses a carefully computed dependence coefficient that quantifies the probabilistic dependence between tuples in a fine-grained manner. We interpret this coefficient as the ratio of dependent indistinguishability of a tuple which is the maximum change in a query output due to the modification of another dependent tuple and self indistinguishability which is the maximum change in a query output due to modification of the tuple itself.

In summary, our work makes the following contributions:

- **Inference Attack:** Using real-world datasets we demonstrate the feasibility of an inference attack on noisy query results generated by existing DP mechanisms through utilizing the dependence between tuples. We show that an adversary can infer sensitive location information about a user from private query outputs by exploiting her social relationships. Furthermore, this adversary, even with partial knowledge of both the user’s social network and the tuple database, can extract more sensitive location information than the adversary that knows all the tuples in the database except one but is unaware of their dependence relationships.

- **Dependent Differential Privacy:** We formalize the notion of DDP, to defend against adversaries who have prior information about the probabilistic dependence between tuples in a statistical database. We then show that it is possible to achieve the DDP guarantees by augmenting the Laplace mechanism, used for achieving the DP guarantees, with a dependence coefficient. The coefficient allows accurate computation of the query sensitivity for dependent data, thus minimizing the noise that needs to be added providing better utility at the
same privacy level. Furthermore, we prove that our dependent perturbation mechanism is also resilient to composition attacks \cite{33,48}.

- **Evaluation:** Our proposed dependent perturbation mechanism applies to any class of query functions. Using extensive evaluation involving different query functions (e.g., machine learning queries such as clustering and classification, and graph queries such as degree distribution) over multiple large-scale real-world datasets we illustrate that our DPM outperforms state-of-the-art approaches in providing rigorous privacy and utility guarantees for dependent tuples.

The frequently used mathematical notations used in this chapter are listed in Table 4.1.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mathcal{D}$</td>
<td>Random variable for the input database</td>
</tr>
<tr>
<td>$D_i, D_j'$</td>
<td>Two neighboring databases (instances of $\mathcal{D}$)</td>
</tr>
<tr>
<td>$D_i$</td>
<td>The $i$-th record in the database</td>
</tr>
<tr>
<td>$D_{-i}$</td>
<td>All other records except for $D_i$ in the database</td>
</tr>
<tr>
<td>$Q(\cdot)$</td>
<td>Query function</td>
</tr>
<tr>
<td>$\mathcal{A}(\cdot)$</td>
<td>Privacy-preserving mechanism</td>
</tr>
<tr>
<td>$S$</td>
<td>Noisy output</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Privacy parameter</td>
</tr>
<tr>
<td>$\hat{D}_i$</td>
<td>The estimated value of $D_i$ by the adversary</td>
</tr>
<tr>
<td>$L$</td>
<td>The number of correlated tuples in the database</td>
</tr>
<tr>
<td>$R$</td>
<td>The probabilistic dependence relationship among tuples</td>
</tr>
<tr>
<td>$H(\cdot)$</td>
<td>The entropy of a variable</td>
</tr>
<tr>
<td>$\mu$</td>
<td>The true centroids of the K-means algorithms</td>
</tr>
<tr>
<td>$\hat{\mu}$</td>
<td>The differentially private centroids of the K-means algorithms</td>
</tr>
<tr>
<td>$\hat{\mu}$</td>
<td>The estimated centroids of the K-means algorithms</td>
</tr>
<tr>
<td>$\rho_{ij}$</td>
<td>Dependence coefficient between $D_i$ and $D_j$</td>
</tr>
<tr>
<td>$DS_{ij}^Q$</td>
<td>Dependent sensitivity of $Q$ over $D_j$ induced by the modification of $D_i$</td>
</tr>
</tbody>
</table>
4.2 Adversarial Model

The popularity of DP as a privacy definition (recall Definition 1) stems from the fact that it makes no assumptions about the background knowledge available to an adversary. In other words, mechanisms that satisfy the DP definition, guarantee that users’ sensitive data are protected regardless of adversarial knowledge. However, the existing DP frameworks do not explicitly model tuple dependence/correlation in the database [22, 57, 71, 72, 81, 139]. In reality, this may leave open the possibilities of inference attacks as data from different users can be dependent, where the dependence can be due to various social, behavioral and genetic interactions that might exist between users. An active adversary can use auxiliary information channels to access these dependence and exploit the vulnerabilities in existing DP mechanisms as illustrated by the simple example below.

Example 1: Consider a database $D = [D_i, D_j]$ where $D_i, D_j$ have a probabilistic dependence as $D_j = 0.5D_i + 0.5X$ and $D_i, X$ are independently and uniformly distributed within $[0, 1]$ as shown in Fig. 4.1. Below we consider a simple inference attack in which an adversary issues a sum query $Q(D) = D_i + D_j$ and uses the query result to infer the value of $D_i$.

First, following the causal perspective in [121] and using the LPM mechanism, we...
add a Laplace noise with parameter \(1/\epsilon\) to the query output. Thus, we can derive the privacy guarantee of \(D_i\) as
\[
\max_S \frac{P(A([D_i=0,D_j]=S))}{P(A([D_i=1,D_j]=S))} \leq \exp(\epsilon).
\]

Next, we follow the associative perspective in [121] that explicitly models the dependence/correlation existing between \(D_i\) and \(D_j\). Under the same amount of Laplace noise with parameter \(1/\epsilon\), we have the privacy guarantee of \(D_i\) as
\[
\max_S \frac{P(A([D_i=0,D_j]=S))}{P(A([D_i=1,D_j]=S))} = \max_S \int_{-\infty}^{\infty} \exp\left(-\frac{|S-0-D_i|}{\epsilon}\right) \leq \exp(1.5\epsilon). \tag{1}
\]
Therefore, the participation of \(D_j\) would result in a privacy degradation of \(D_i\) caused by the dependence/correlation relationship between them.

The above example motivates us to consider the DDP-adversary which explicitly models the dependence/correlation among tuples defined as below.

**DDP-adversary:** We assume a setting, in which a trusted data curator maintains a statistical database \(D = [D_1, D_2, \cdots, D_n]\) where \(D_i\) denotes the data from the \(i^{th}\) user. In response to a query, the curator computes a randomized query result \(A(D)\), with the goal of providing statistical information about the dataset while preserving the privacy of individual users. Both the users and the data curator are assumed to be honest. The data recipient, issuing the query, is our DDP-adversary. We associate the following properties to this adversary who wants to use the noisy query result to infer data \(D_i\):

- **Access to** \(\mathbb{D}_{-i}\): Data of all the other \(n - 1\) users (excluding the \(i^{th}\) user), denoted by \(\mathbb{D}_{-i}\), is available to the adversary. This property makes the DDP-adversary as powerful as an i.i.d.-adversary.

- **Access to joint distribution** \(P(D_1, \ldots, D_n)\): The adversary uses auxiliary channels (e.g., the Gowalla social network in our attack in Section 4.3) to estimate the joint probability distribution \(P(D_1, \ldots, D_n)\), between the data tuples. This prop-
ery together with access to $D_{-i}$ makes a DDP-adversary more powerful than an i.i.d.-adversary.

In the remaining work, unless otherwise specified, the privacy definitions and guarantees are all with respect to the DDP-adversary. In Section 4.3, we perform a real-world inference attack to demonstrate that a DDP-adversary can extract more private information than guaranteed by DP. In Section 4.4, we develop a new privacy definition dependent differential privacy (DDP) that allows for dependence between data tuples in the database. In Section 4.5, we propose a privacy mechanism to satisfy the DDP definition. We establish formal guarantees for our privacy mechanism and illustrate its efficacy using experiments on large-scale real-world datasets.

4.3 Inference Attack: Differential Privacy under Dependent Tuples

Real-world datasets are complex networks that exhibit strong dependence (correlations) and their release introduces various privacy challenges. Adversaries can combine the released obfuscated data (generated by applying the privacy mechanisms on the data), with knowledge of the existing dependence relations to infer users’ sensitive information. There exist limited prior work that have outlined realistic inference attacks exposing the vulnerability of existing DP mechanisms under dependent data tuples [71,72]. In this section, we demonstrate (1) a real-world inference attack on the LPM-based differential privacy mechanism, as a realistic confirmation of the feasibility of such attacks in practical scenarios; and (2) the capability of a DDP-adversary to use released data, generated by exiting DP mechanisms, to build an inference attack which violates the security guarantees of existing DP mechanisms. Before out-
lining our real inference attack for DP we compare our work with existing related
work \cite{46,48,71} to highlight the importance of our attack.

- Ganta et al. in \cite{48} explored how one can reason about privacy in the presence of
  independent anonymized releases of overlapping data. Compared with our inference
  attack, they do not consider the dependence between data tuples in their attack.

- Fredrikson et al. in \cite{46} considered predicting a patient’s genetic marker from
  the differentially private query results by utilizing demographic information about
  that patient. Thus, the auxiliary information used in this attack is additional
  information about a patient (single tuple) and not dependence between tuples.

- Kifer et al. in \cite{71} investigated the inference about the participation of an edge
  in a social network through observing the number of inter-community edges. The
  inference performance varied with different network generation models. In contrast
  to the theoretical work of Kifer et al., we demonstrate inference attacks using real
  data on complex differentially private machine learning queries.

4.3.1 Dataset Description

The increasing exposure of user’s location information has started raising serious
privacy concerns \cite{11,112,113}. Not only the location information itself is sensitive,
but more seriously it can be easily linked to a variety of other information that an
individual user usually wishes to protect such as her home, work location, political
views, etc.

Sharing of location information is often associated with serious privacy threats
\cite{11,112,113}. Location data (or mobility traces), can be easily linked with auxiliary
information sources (such as maps) to not only infer places such as home and work
location but also a user’s political views, her medical conditions, etc.

We use the data collected from the location-based social networking service Gowalla [24] for mounting our attack to infer user’s location information. The locations correspond to users’ check-ins at places. We obtained a subset of their location dataset which had a total of 196,591 users and 6,442,890 check-ins of these users over the period from February 2009 to October 2010. Gowalla also maintains an associated social network connecting its users. In fact, it is this correlation that forms the basis of our inference attack. The network data we obtained consisted of 950,327 edges connecting 196,591 users. Considering the sparsity of the location information, we decided to restrict our analysis to users around three cities: New York, San Francisco and Los Angeles. We selected these cities since they had the highest number of active users. For our attack, we used data from users who performed at least 10 check-ins at locations within a 25km radius in any of the three cities. The resulting dataset contains 6,969 users, 98,802 check-ins and 17,770 locations. The corresponding selected social dataset contains 47,502 edges connecting these 6,969 users.

**Constructing the Location Pattern Dataset:** For each user $i$, we collect her check-ins in a set $C(i) = \{c_{i0}, c_{i1}, c_{i2}, \ldots, c_{im_i}\}$, where each check-in $c_{ik} = \text{[User ID, timestamp, GPS coordinates, place identifier]}$. The GPS coordinates $= [lat, lon]$ represents the latitude and longitude of the location shared by the user. For the inference attack, we only consider the GPS coordinates as effective check-in records, and use them to extract a location pattern vector for each user. To do so we compute the frequency of visits to each location, and only keep the latitude and longitude of those locations that correspond to the top-$q$ frequencies. Formally, the location pattern vector for user $i$ is defined as:

$$d_i = (\text{lat}_{i1}, \text{lon}_{i1}, \text{lat}_{i2}, \text{lon}_{i2}, \text{lat}_{i3}, \text{lon}_{i3}, \ldots, \text{lon}_{iq})$$ (4.1)
where \((\text{lat}_{i_1}, \text{lon}_{i_1})\) is the coordinate of the most frequently visited location of user \(i\), \((\text{lat}_{i_2}, \text{lon}_{i_2})\) and \((\text{lat}_{i_3}, \text{lon}_{i_3})\) correspond to the second and the third most frequently visited location of user \(i\), respectively. Without loss of generality, we normalize each attribute in the location pattern dataset such that its value lies within \([0, 1]\).

### 4.3.2 Differentially Private Data Release

The GPS coordinates describing a user’s check-ins are generally considered to be distinct, but in reality they are typically clustered around a limited number of points. We consider a scenario where the data provider uses the classical K-means clustering approach [54] on the Gowalla location dataset to compute cluster centroids, applies DP mechanism on the centroids, and publishes the perturbed centroids to other applications or researchers.

#### 4.3.2.1 K-means Clustering

The input to the K-means algorithm are points \(d_1 \ldots d_n\) in the \(2q\)-dimensional unit cube \([0, 1]^{2q}\). For our attack, we choose \(q = 6\) while constructing the location pattern in Eq. 4.1 To preserve the privacy of users’ sensitive data, the data provider perturbs the true centroids \(\mu = (\mu_1, \mu_2, \cdots, \mu_k)\) by using the LPM mechanism, and releases the perturbed centroids \(\tilde{\mu} = (\tilde{\mu}_1, \tilde{\mu}_2, \cdots, \tilde{\mu}_k)\) for preserving the privacy of each individual’s location pattern. In this work, we use \(\tilde{a}\) to represent the perturbed version of \(a\), and \(\hat{a}\) to represent the estimated value of \(a\).

Fig. 4.2 depicts the structure of the Gowalla social network dataset which is colored according to the K-means clustering results of the Gowalla location dataset (users belonging to the same community are giving the same color). We can see that
Figure 4.2. **Gowalla’s social dataset colored according to the results from K-means clustering of the location dataset.** We can see that the location dataset is inherently correlated and the social dataset well represents such relationships.

Figure 4.3. **The distance between the location vectors of users.**

Users’ location patterns are inherently correlated, and the social dataset embeds relationships contained in the location dataset. Fig. 4.3 further shows the cumulative distribution function of the location pattern distance between different user pairs, where we compute the distance for $\mathbf{d}_i$ and $\mathbf{d}_j$ as

$$\text{Dist}(\mathbf{d}_i, \mathbf{d}_j) = \frac{1}{q} \sum_{l=1}^{q} \text{dist} ((\text{lat}_{il}, \text{lon}_{il}), (\text{lat}_{jl}, \text{lon}_{jl}))$$

(4.2)
and \( \text{dist}((\text{lat}_d, \text{lon}_d), (\text{lat}_d', \text{lon}_d')) \) represents the earth’s surface distance between two coordinates\(^2\).

We find that the distance between the location patterns for closer friends is smaller. These observations from Fig. 4.2 and Fig. 4.3 not only imply that the location patterns of users are correlated with each other, but also that their social network can serve as an important external information source for an adversary to infer a user’s sensitive records.

### 4.3.3 Inference Algorithm

The adversary can observe the differentially private community centroids \( \hat{\mu} = [\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_k] \), has access to auxiliary information \( D_{-i} \) (recall DDP-adversary in Section 4.2) and also to the social relationships among the users. Let the adversary’s estimated value of \( D_i \) be \( \hat{D}_i \). Using Bayes’ theorem, the posterior probability of \( \hat{D}_i = \hat{d}_i \) computed by the DDP-adversary can thus be written as

\[
P(\hat{D}_i = \hat{d}_i | \hat{\mu}, D_{-i}) = \frac{P(\hat{\mu}, D_{-i} | \hat{D}_i = \hat{d}_i) P(\hat{D}_i = \hat{d}_i)}{P(\hat{\mu}, D_{-i})} \\
= \frac{P(\hat{\mu} | D_{-i}, \hat{D}_i = \hat{d}_i) P(D_{-i} | \hat{D}_i = \hat{d}_i) P(\hat{D}_i = \hat{d}_i)}{P(\hat{\mu} | D_{-i})} \\
= \frac{P(\hat{\mu} | D_{-i}, \hat{D}_i = \hat{d}_i) P(D_{-i} | \hat{D}_i = \hat{d}_i) P(\hat{D}_i = \hat{d}_i)}{P(\hat{\mu} | D_{-i})} \\
\sim \exp \{-|\hat{\mu} - \hat{\mu}| \epsilon\} \cdot P(D_{-i} | \hat{D}_i = \hat{d}_i) \tag{4.3}
\]

where \( \exp \{-|\hat{\mu} - \hat{\mu}| \epsilon\} \) in Eq. 4.3 represents the Laplace noise induced by the estimated centroids \( \hat{\mu} \). Such estimated centroids are computed using the auxiliary \( D_{-i} \) of other users and each potential value \( \hat{D}_i = \hat{d}_i \), \( P(\hat{D}_i = \hat{d}_i | D_{-i}) \) represents the prior.

\(^2\)http://www.movable-type.co.uk/scripts/latlong.html
information of $D_i$ inferred from the auxiliary information $\mathbb{D}_{-i}$ of other users. Note that our inference attack can be mounted with any amount of auxiliary information. For an adversary with partial auxiliary information, the corresponding estimated centroids $\hat{\mu}$ and prior information $P(\hat{D}_i = \hat{d}_i|\mathbb{D}_{-i})$ are computed based on these partial auxiliary information.

To estimate $\hat{D}_i$, an adversary can discretize the potential region of $D_i$ and compute

$$\exp\{-|\hat{\mu} - \hat{\mu}|\epsilon\} \cdot P(\hat{D}_i = \hat{d}_i|\mathbb{D}_{-i})$$

for each potential value $\hat{d}_i$. The adversary can then estimate $\hat{D}_i$ which corresponds to the maximal posterior probability for all the potential values $\hat{d}_i$, i.e.,

$$\hat{D}_i = \arg\max_{\hat{d}_i} \exp\{-|\hat{\mu} - \hat{\mu}|\epsilon\} \cdot P(\hat{D}_i = \hat{d}_i|\mathbb{D}_{-i})$$

(4.4)

The key challenge is for the adversary to compute the prior information $P(\hat{D}_i = \hat{d}_i|\mathbb{D}_{-i})$. We consider two different types of adversaries: one which assumes that the tuples are independent and the other which utilizes the social relationships between the users.

![Figure 4.4. Location vector inference attack. (a) represents Attack 1 under independent tuple assumption and (b) represents Attack 2 under dependent tuple assumption.](image-url)
4.3.3.1 Attack 1 (Without Leveraging Dependence/Correlation)

First, we consider an adversary who does not leverage dependence/correlation across users in the inference attacks. To simplify our analysis without loss of generality, we assume that \( D_i \) is independent of \( \{D_j\}_{j=0, j\neq i}^{n-1} \) (i.e., \( P(\hat{D}_i = \hat{d}_i | D_{-i}) = P(\hat{D}_i = \hat{d}_i) \)) with identical distributions. Therefore, the auxiliary information \( D_{-i} \) can serve as sampling values of \( D_i \), which can be utilized to estimate the prior probability \( P(\hat{D}_i = \hat{d}_i) \).

Fig. 4.4 (a) shows the mechanism for inference attack under the independent assumption. We discretize the estimated region for \( \hat{D}_i \) where each grid corresponds to a potential value \( \hat{d}_i \) of \( D_i \). The red squares (friends of user \( i \)) and the green triangles (non-friends of user \( i \)) are location patterns of the other users which also represent the sampling values of \( D_i \). Based on these sampling values, we estimate the prior probability of \( \hat{D}_i = \hat{d}_i \) by counting the number of values in \( D_{-i} \) that fall into the grid of \( \hat{d}_i \) as

\[
P_{\text{inde}}(\hat{D}_i = \hat{d}_i | D_{-i}) = \frac{|d_j : d_j \in \text{grid}(\hat{d}_i)|}{\sum_{d_k} |d_j : d_j \in \text{grid}(d_k)|}
\]

(4.5)

4.3.3.2 Attack 2 (Leveraging Dependence/Correlation)

Next, we consider a sophisticated adversary who aims to leverage the dependence/correlation across users in the inference attacks. Such an assumption is practical since the mobility traces from close friends are likely to be similar as shown in Fig. 4.3. For an adversary who has access to the social relationships of the users, he can draw circles, shown in red, in Fig. 4.4(b), to represent the dependent relationships among users, and all the girds (corresponding to each potential value \( \hat{d}_i \)) within the red circles would be given a higher weight. The prior probability for
\( \hat{D}_i = \hat{d}_i \) would thus be weighted based on the relationships of the users, and the weighted prior probability under the dependent assumption would become

\[
P_{de}(\hat{D}_i = \hat{d}_i | \mathbb{D}_{-i}) = \frac{\sum_{d_j \in \text{grid}(\hat{d}_i)} \text{weight}(\hat{d}_i)}{\sum_{d_j \in \text{grid}(\hat{d}_j)} \text{weight}(\hat{d}_j)}
\]  

(4.6)

In Fig. 4.4(a), we can see that there are three sampling values that belong to the grids corresponding to \( \hat{D}_i = \hat{d}_{i(4)} \) and \( \hat{D}_i = \hat{d}_{i(8)} \). Therefore, we have \( P_{inde}(\hat{D}_i = \hat{d}_{i(4)} | \mathbb{D}_{-i}) = P_{inde}(\hat{D}_i = \hat{d}_{i(8)} | \mathbb{D}_{-i}) \). However, in Fig. 4.4(b), the grid for \( \hat{D}_i = \hat{d}_{i(4)} \) would have a much higher weight than the grid for \( \hat{D}_i = \hat{d}_{i(8)} \). Therefore, we have \( P_{de}(\hat{D}_i = \hat{d}_{i(4)} | \mathbb{D}_{-i}) > P_{de}(\hat{D}_i = \hat{d}_{i(8)} | \mathbb{D}_{-i}) \). As we know the location patterns of the user \( i \)'s friends (shown as the red squares Fig. 4.4), it is more likely that the location pattern \( D_i \) of user \( i \) will be located closer to her friends based on our observations in Fig. 4.3.

![Figure 4.5. The inference error of the adversary’s inference attack.](image)

We evaluate the performance for these two inference attacks by measuring the following metric of inference error:

\[
\text{Inference Error} = \frac{1}{n} \sum_{i=1}^{n} \text{Dist}(d_i, \hat{D}_i)
\]  

(4.7)

where \( \text{Dist}(\cdot) \) is defined in Eq. 4.2.
We set the number of communities $K = 8$ in the K-means algorithm (shown in Fig. 4.2) and discretize each city (NY, SF, LA) into $20 \times 20$ grids. From the result in Fig. 4.5, we can see that the attacker can exploit the social relationships between users to make better inferences (shown by smaller inference errors in Fig. 4.5). Furthermore, in Fig. 4.5, a larger $\epsilon$ (worse privacy guarantee) results in smaller inference errors, since the adversary has access to more accurate centroids.

From our analysis above, we can see that an adversary that aims to clearly model and leverage the dependence/correlation across tuples could infer more information than expected. Note that we used the location data just as an example, and our attack observations are broadly applicable to any dataset that exhibits probabilistic dependence between user records. Therefore, from the associative perspective in [121], we have to take the dependence/correlation relationships into consideration when protecting privacy for real-world dependent tuples.

### 4.4 Dependent Differential Privacy

As demonstrated in Section 4.3, existing DP mechanisms underestimates the privacy risk in the presence of dependent tuples, resulting in degradation of expected privacy. Hence, a new privacy notion is required to explicitly incorporate the dependence/correlation relationship existing among tuples.

Recent work has made attempts to capture and model this notion of tuple dependence and correlation in databases. The Pufferfish framework [72], proposed as a generalization of DP, incorporates adversarial belief about a database and its generation as a distribution over all possible database instances. The Blowfish framework [57], which is a subclass of the Pufferfish framework, allows a user to specify adversarial knowledge about the database in the form of deterministic policy constraints.
Motivated by the above frameworks, we formalize the notion of dependent differential privacy, as a subclass of the general Pufferfish framework, incorporating probabilistic dependence between the tuples in a statistical database. In addition, we also propose an effective perturbation mechanism (Section 4.5) that can provide rigorous privacy guarantees. In contrast, there are no general algorithms known for achieving Pufferfish privacy.

For any database \( D = [D_1, D_2, \cdots, D_n] \), we define its dependence size to be \( L \) if any tuple in \( D \) is dependent on at most \( L - 1 \) other tuples. \( L \) is domain-specific and we refer interested readers to [22] for more explanations of \( L \). We denote by \( \mathcal{R} \) the probabilistic dependence relationship among the \( L \) dependent tuples. Relationship \( \mathcal{R} \) could be due to the data generating process as specified in [71] or could be due to other social, behavioral and genetic relationships arising in real-world scenarios. We provide an instance of \( \mathcal{R} \) in Section 4.3, where dependence in the Gowalla location dataset was introduced via the Gowalla social network dataset and such dependence is probabilistic instead of deterministic as in Blowfish framework [57]. The DDP framework is equivalent to the DP framework when \( \mathcal{R} \) represents independence between data tuples. We begin by defining the dependent neighboring databases as follows:

**Definition 8.** Two databases \( D(L, \mathcal{R}), D'(L, \mathcal{R}) \) are dependent neighboring databases, if the modification of a tuple value in database \( D(L, \mathcal{R}) \) (e.g., the change from \( D_i \) in \( D(L, \mathcal{R}) \) to \( D'_i \)) causes change in at most \( L - 1 \) other tuple values in \( D'(L, \mathcal{R}) \) due to the probabilistic dependence relationship \( \mathcal{R} \) between the data tuples.

Based on the above dependent neighboring databases, we define our dependent differential privacy as follows.

**Definition 9.** (\( \epsilon \)-Dependent Differential Privacy) A randomized algorithm \( A \) provides \( \epsilon \)-dependent differential privacy, if for any pair of dependent neighboring databases
\(D(L, \mathcal{R})\) and \(D'(L, \mathcal{R})\) and any possible output \(S\), we have

\[
\max_{D(L, \mathcal{R}), D'(L, \mathcal{R})} \frac{P(\mathcal{A}(D(L, \mathcal{R})) = S)}{P(\mathcal{A}(D'(L, \mathcal{R})) = S)} \leq \exp(\epsilon)
\]

(4.8)

where \(L\) denotes the dependence size and \(\mathcal{R}\) is the probabilistic dependence relationship between the data tuples.

From Definition 9, we see that dependent differential privacy restricts an adversary’s ability to infer the sensitive information of an individual tuple, even if the adversary has complete knowledge of the probabilistic dependence relationship \(\mathcal{R}\) between the tuples.

### 4.4.1 Security Analysis

Dinur et al. [33] proved that unless a particular amount of noise is added to the query responses, an adversary can use a polynomial number of queries to completely reconstruct the database. Therefore, any privacy framework must provide privacy guarantees for multiple queries, in order to defend against such composition attacks [33, 48].

In the following, we show that DDP is secure against these composition attacks. Here, ‘secure’ means that the algorithms that provide strict DDP also provide meaningful privacy in the presence of auxiliary information. To this end, we propose both the sequential composition theorem and the parallel composition theorem for DDP by extending the previous results on composition for DP in [90]. Our analysis show that the composition properties for DDP provide privacy guarantees in a well-controlled manner, rather than collapsing rapidly as other approaches in [48]. The proofs for Theorems 9 and 10 follow directly from the ones presented in [90] for differential privacy and are deferred to B.1 in the Appendix to improve readability.
Sequential Composition Theorem Multiple queries that each provides dependent differential privacy in isolation provide dependent differential privacy in sequence.

**Theorem 9.** Let randomized algorithm $A_t$ each provide $\epsilon_t$-dependent differential privacy under the dependence size $L$ and probabilistic dependence relationship $R$ over the same input data $D$. The sequence of these algorithms $A_t$ provides $\sum_t \epsilon_t$-dependent differential privacy under the same $L, R$.

Parallel Composition Theorem When the queries are applied to disjoint subsets of the data, we have the parallel composition theorem as

**Theorem 10.** Let randomized algorithms $A_t$ provide $\epsilon_t$-dependent differential privacy under the dependence size $L$ and probabilistic dependence relationship $R$. We denote by $D_t$ the arbitrary disjoint subsets of the input domain $D$. The sequence of these randomized algorithm $A_t$ provides $\max_t \epsilon_t$-dependent differential privacy under the same $L, R$.

### 4.4.2 Privacy Axioms

Kifer et al. in [70] suggested two privacy axioms: *transformation invariance* and *convexity* that should be satisfied by any consistent privacy definition. The following theorems show that our DDP satisfies both the axioms.

**Theorem 11.** Transformation Invariance Property: For a randomization algorithm $A$ that satisfies $\epsilon$-dependent differential privacy under the dependence size $L$ and probabilistic dependence relationship $R$ and any other randomization algorithm $B$, $B(A(\cdot)) = B(A(\cdot))$ also satisfies $\epsilon$-dependent differential privacy under the same $L, R$.

**Theorem 12.** Convexity Property: For two randomization algorithms $A_1, A_2$ that both satisfy $\epsilon$-dependent differential privacy under the dependence size $L$ and probabilistic dependence relationship $R$, let $A^p$ represent an algorithm that runs $A_1$ with
probability $p$ and runs $A_2$ with probability $1 - p$, then $A^p$ also satisfies $\epsilon$-dependent differential privacy under the same $L, R$.

Proofs for the above two theorems are also deferred to B.1 in the Appendix to improve readability.

\[
\max_{d_{i1}, d_{i2}} \frac{P(A([D_i = d_{i1}, D_j]) = [\tilde{d}_i, \tilde{d}_j])}{P(A(D_i = [d_{i2}, D_j]) = [\tilde{d}_i, \tilde{d}_j])} = \frac{P(\tilde{D}_i = \tilde{d}_i | D_i = d_{i1}) \sum_{d_j} P(D_j = d_j | D_i = d_{i1}) P(\tilde{D}_j = \tilde{d}_j | D_j = d_j)}{P(\tilde{D}_i = \tilde{d}_i | D_i = d_{i2}) \sum_{d_j} P(D_j = d_j | D_i = d_{i2}) P(\tilde{D}_j = \tilde{d}_j | D_j = d_j)} \leq \max_{d_{i1}, d_{i2}} \frac{P(\tilde{D}_i = \tilde{d}_i | D_i = d_{i1}) \sum_{d_j} P(D_j = d_j | D_i = d_{i1}) P(\tilde{D}_j = \tilde{d}_j | D_j = d_j)}{P(\tilde{D}_i = \tilde{d}_i | D_i = d_{i2}) \sum_{d_j} P(D_j = d_j | D_i = d_{i2}) P(\tilde{D}_j = \tilde{d}_j | D_j = d_j)}
\]

(4.12)

4.5 Mechanism Design for DDP

In this section, we design an effective mechanism to achieve $\epsilon$-dependent differential privacy and support private query results over dependent tuples. We also describe extensions to the existing LPM-based differential privacy scheme that allows it to be used in the DDP setting.

It is interesting to find that the LPM-based differential privacy scheme could be tuned to provide provable privacy guarantees even in the dependent setting by inducing an extra parameter dependent coefficient, which measures the extent of tuples’ dependence relationship. Comparing with an existing approaches, our privacy mechanism shows significant privacy and utility superiority.
To provide more insights into our privacy mechanism design, we take a further look at Example 1 in Section 4.2. Recall that the probabilistic dependence relationship \( R \) was specified as \( D_j = 0.5D_i + 0.5X \) where \( D_i, X \) are independently and uniformly distributed within \([0, 1]\).

Quantifying the performance of LPM: We use separation as a metric to analyze the performance of the LPM-based differential privacy scheme under dependent tuples. Separation measures the maximum difference between two Laplace distributions of \( P(A([D_i = d_1, D_j]) = S) \) and \( P(A([D_i = d_2, D_j]) = S) \). Smaller separation implies better privacy performance. We further consider three query functions sum, subtraction, multiplication over the same dependent database. To achieve DP, an Laplace noise with parameter \( 1/\epsilon \) is added to each of the three query results. Assuming independent tuples, the separation for each noisy query output is the same \( \epsilon \) as guaranteed by DP (shown in Fig. 4.6). In comparison, under the probabilistic dependence between tuples, we have the following interesting observations: 1) the separation may become larger for the dependent tuples than that under the independent assumption (see the sum query in Fig. 4.6); and 2) the change of separation caused by the same dependent database may vary for different queries. In this section, we aim to develop a principled perturbation mechanism for supporting arbitrary query
functions, by introducing an extra parameter *dependence coefficient* to measure the fine-grained dependence relationship between tuples.

### 4.5.1 Baseline Approach

A database $D(L, R)$ with dependence size $L$ would result in a quicker exhaustion of the privacy budget $\epsilon$ in DP by a factor of $L$. This observation provides the baseline approach for achieving the $\epsilon$-dependent differential privacy as stated in the theorem below:

**Theorem 13.** An $\epsilon/L$-differentially private mechanism $A(D) = Q(D) + \text{Lap}(L\Delta Q/\epsilon)$ over a database $D$ with the dependence size $L$ achieves $\epsilon$-dependent differential privacy, for a query function $Q$ with global sensitivity $\Delta Q$.

While the above theorem follows directly from the definition of DDP, the baseline approach is not optimal as it implicitly assumes that all the dependent tuples in the database are *completely dependent* on each other. By *completely dependent*, we mean that the change in one tuple would cause a dependent tuple to change by the maximum domain value, thus making the sensitivity of the query over the two tuples twice the sensitivity under the independent assumption. As we can see from Fig. 4.6, the sensitivity for the *sum* query $\Delta Q$, under the independent tuple assumption, is 1 as $D_i \in [0, 1]$. Under the dependent tuples, the maximum change in $D_j$ caused by the change of $D_i$ is 0.5, which is only half of the maximum domain value for $D_j$. Therefore, the sensitivity of the *sum* query over the two dependent tuples is $1.5$, which is smaller than $2 \times \Delta Q = 2$ as considered in the baseline approach.

This conservative assumption of *completely dependent* tuples results in the addition of a lot of unnecessary noise to the query output rendering it unusable. In real-world datasets, although the tuples are related, only a few of them are *completely*
dependent on each other. This insight motivates us to explore mechanisms that can use less amount of noise but still satisfy all the guarantees provided by \( \epsilon \)-dependent differential privacy.

### 4.5.2 Our Dependent Perturbation Mechanism

To minimize the amount of added noise we want to identify the fine-grained dependence relationship between tuples and use it to design the mechanism. We begin with a simple query function (e.g., an identity query) over a dataset with only two tuples \( D = [D_i, D_j] \). The privacy objective is to publish a sanitized version of the dataset i.e., \( \tilde{D} = [\tilde{D}_i, \tilde{D}_j] \) as query output. We later generalize our analysis to scenarios involving arbitrary query functions over databases with more than two tuples, i.e., \( D = [D_i, D_j, D_k, \ldots] \). According to Definition 9 to satisfy \( \epsilon \)-dependent differentially privacy requires

\[
\max_{d_i_1, d_i_2} \frac{P \left( \mathcal{A}([D_i = d_{i_1}, D_j]) = [\tilde{d}_{i_1}, \tilde{d}_{i_2}] \right)}{P \left( \mathcal{A}([D_i = d_{i_2}, D_j]) = [\tilde{d}_{i_1}, \tilde{d}_{i_2}] \right)} \leq \exp(\epsilon)
\]  

(4.11)

where the output distributions of \( \mathcal{A} \), due to the change in \( D_i \) from \( d_{i_1} \) to \( d_{i_2} \), would be bounded.

Motivated by the LPM in Section 2.2.1 we continue to use Laplace noise for perturbing the true query output to satisfy \( \epsilon \)-dependent differential privacy. Our objective thus reduces to finding a proper scaling factor \( \sigma(\epsilon) \) for the required Laplace distribution. According to the law of total probability, we further transform the left-handside (LHS) of Eq. 4.11 to Eq. 4.12. We restrict ourselves to discrete variables for simplicity, but all the results will also apply to the continuous case as in [6]. For the
first term of the right-handside (RHS) of Eq. 4.12, we have

$$\max_{\tilde{d}_i, \tilde{d}_i} P(\tilde{D}_i = \tilde{d}_i | D_i = d_{i_1}) = \max_{\tilde{d}_i, \tilde{d}_i} \frac{\exp \left( \frac{\|\tilde{d}_i - d_{i_1}\|_1}{\sigma(\epsilon)} \right)}{\exp \left( - \frac{\|\tilde{d}_i - d_{i_2}\|_1}{\sigma(\epsilon)} \right)}$$

$$\leq \max_{\tilde{d}_i, \tilde{d}_i} \exp \left( \frac{\|d_{i_1} - d_{i_2}\|_1}{\sigma(\epsilon)} \right)$$

$$\leq \exp \left( \frac{\Delta D_i}{\sigma(\epsilon)} \right)$$

(4.13)

where $\Delta D_i$ is the maximal difference due to the change in $D_i$. If we ignore the second term in the RHS of Eq. 4.12 and combine the remaining terms with Eq. 4.11 and Eq. 4.13 we obtain the scaling factor of the Laplace noise as $\sigma(\epsilon) = \frac{\Delta D_i}{\epsilon}$, which is exactly the same form as in traditional DP [34]. Therefore, the LPM that satisfies DP is only a special case for our mechanism. The second term in the RHS of Eq. 4.12 incorporates the dependence relationship between $D_i, D_j$ and we will focus our study on this term.

To evaluate the extent of dependence induced in $D_j$ by the modification of $D_i$, we define the dependence coefficient $\rho_{ij}$ as

$$\exp \left( \frac{\rho_{ij} \Delta D_j}{\sigma(\epsilon)} \right)$$

$$= \sum_{d_j} P(D_j = d_j | D_i = d_{i_1}) P(\tilde{D}_j = \tilde{d}_j | D_j = d_j)$$

$$= \sum_{d_j} P(D_j = d_j | D_i = d_{i_2}) P(\tilde{D}_j = \tilde{d}_j | D_j = d_j)$$

(4.14)
Next, we aim to prove that $0 \leq \rho_{ij} \leq 1$. We first have,

$$\frac{\sum_{d_j} P(D_j = d_j | D_i = d_{i_1}) P(\tilde{D}_j = \tilde{d}_j | D_j = d_j)}{\sum_{d_j} P(D_j = d_j | D_i = d_{i_2}) P(\tilde{D}_j = \tilde{d}_j | D_j = d_j)} = \max_{d_{i_1}, d_{i_2}} \frac{\sum_{d_j} P(D_j = d_j | D_i = d_{i_1}) \exp \left( -\frac{\|\tilde{d}_j - d_j\|_1}{\sigma(\epsilon)} \right)}{\sum_{d_j} P(D_j = d_j | D_i = d_{i_2}) \exp \left( -\frac{\|\tilde{d}_j - d_j\|_1}{\sigma(\epsilon)} \right)} \leq \max_{d_{i_1}, d_{i_2}} \exp \left( \frac{\|d_j - d_{j}^{\min}\|_1}{\sigma(\epsilon)} \right) \leq \exp \left( \frac{\Delta D_j}{\sigma(\epsilon)} \right)$$

(4.15)

where $d_j^{\min}$ is the value of $d_j$ that minimizes $\exp \left( \frac{\|d_j - d_{j}^{\min}\|_1}{\sigma(\epsilon)} \right)$. Comparing Eq. 4.14 and Eq. 4.15 we have $\rho_{ij} \leq 1$. Furthermore, it is obvious that

$$\frac{\sum_{d_j} P(D_j = d_j | D_i = d_{i_1}) P(\tilde{D}_j = \tilde{d}_j | D_j = d_j)}{\sum_{d_j} P(D_j = d_j | D_i = d_{i_2}) P(\tilde{D}_j = \tilde{d}_j | D_j = d_j)} \geq 1$$

(4.16)

Comparing Eq. 4.14 and Eq. 4.15, we have $\rho_{ij} \geq 0$. Finally, combining Eq. 4.11–4.14 we have

$$\max_{d_{i_1}, d_{i_2}} \frac{P(A([D_i = d_{i_1}, D_j]) = [\tilde{d}_i, \tilde{d}_j])}{P(A([D_i = d_{i_2}, D_j]) = [d_i, d_j])} \leq \exp \left( \frac{\Delta D_i}{\sigma(\epsilon)} \right) \exp \left( \frac{\rho_{ij} \Delta D_j}{\sigma(\epsilon)} \right)$$

(4.17)

$$= \exp \left( \frac{(\Delta D_i + \rho_{ij} \Delta D_j)}{\sigma(\epsilon)} \right)$$

Therefore, the sensitivity under the dependence relationship between $D_i$ and $D_j$ can be computed as $\Delta D_i + \rho_{ij} \Delta D_j$.

The dependence coefficient $\rho_{ij} \in [0, 1]$ serves as an effective metric to evaluate the dependence relationship between two tuples in a fine-grained manner. We make the following observations about $\rho_{ij}$:
• $\rho_{ij}$ evaluates the dependence relationship between $D_i$ and $D_j$ from the privacy perspective.

• $\rho_{ij} = 0$ corresponds to the setting where $P(D_j = d_j|D_i = d_i)$ is independent of $d_i$. Therefore, the mechanism that satisfies DP is just a special case of our analysis that takes arbitrary dependence relationship between tuples into consideration. In addition, using the sensitivity definition $\Delta D_i + \rho_{ij} \Delta D_j$, we observe that more noise needs to be added than under the independent assumption that computes the sensitivity as $\Delta D_i$.

• $\rho_{ij} = 1$ corresponds to the completely dependent setting where $D_j$ can be uniquely determined by $D_i$. The baseline approach in Section 4.5.1 is just a special case of our analysis where all the dependent $L$ tuples are completely dependent on each other. As all practical privacy notions require some assumptions on the allowed distributions, it makes sense to analyze the fine-grained dependence relationship in order to maximize utility under the same privacy requirement. Compared with the baseline approach, less noise would be added for our dependent perturbation mechanism since we consider fine-grained dependence relationship. In real-world scenarios, tuples are related but few of them are completely dependent i.e., $\rho_{ij} < 1$. Therefore, our proposed dependent perturbation mechanism can significantly decrease the added noise compared with the baseline approach.

• $\rho_{ij}$ is asymmetric, i.e., $\rho_{ij} \neq \rho_{ji}$. The reason is that the dependence coefficient evaluates the extent of dependence in $D_j$ induced by $D_i$, which is causal and directional. For example, a celebrity’s participation in a social network is likely to result in the participation of her fans. However, it may not be the case the other way around.
To generalize and derive $\rho_{ij}$ for any output $\tilde{d}_j$, we reformulate $\rho_{ij}$ to avoid the appearance of $\tilde{d}_j$. Specifically, we consider the general dependent relationships of tuples to analyze the second term of the RHS of Eq. 4.12 as

\[
\max_{d_{i1}, d_{i2}} \sum_{d_j} P(D_j = d_j | D_i = d_{i1}) \exp\left(-\frac{\|d_j - d_j\|_1}{\sigma(\epsilon)}\right) \\
\leq \max_{d_{i1}, d_{i2}} \sum_{d_j} P(D_j = d_j | D_i = d_{i2}) \exp\left(-\frac{\|d_j - d_j\|_1}{\sigma(\epsilon)}\right) \\
\leq \max_{d_{i1}} \sum_{d_j} P(D_j = d_j | D_i = d_{i1}) \exp\left(\frac{\|d_j - d_j\|_1}{\sigma(\epsilon)}\right) 
\]

(4.18)

where $d_j^{\min}$ is the value of $d_j$ that minimizes $\exp\left(\frac{\|d_j - d_j\|_1}{\sigma(\epsilon)}\right)$. In order to quantify the dependence coefficient which is applicable for any output value of $\tilde{d}_j$, we further have $\exp\left(\frac{\|d_j - d_j\|_1}{\sigma(\epsilon)}\right) \leq \exp\left(\frac{\|d_j - d_j^*\|_1}{\sigma(\epsilon)}\right)$, where $d_j^*$ maximizes $\|d_j - d_j\|_1$. Substituting $d_j^{\min}$ with $d_j^*$, we obtain the following relationship.

\[
\rho_{ij} = \max_{d_{i1}} \log \left\{ \sum_{d_j} P(D_j = d_j | D_i = d_{i1}) \exp\left(\frac{\|d_j - d_j^*\|_1}{\sigma(\epsilon)}\right) \right\} \sigma(\epsilon) 
\]

(4.19)

where $d_j^*$ is the optimal solution to $\arg \max_{d_{j1}} \|d_j - d_j\|_1$.

**Interpreting $\rho_{ij}$**: To further understand the dependence coefficient in Eq. 4.19, we define the *Self* and *Dependent Indistinguishability* terms as follows:

**Self Indistinguishability**

\[
\text{Self Indistinguishability} = \max_{d_{j1}, d_{j2}} \frac{P(D_j = \tilde{d}_j | D_i = d_{j1})}{P(D_j = \tilde{d}_j | D_i = d_{j2})} \\
= \max_{d_{j1}, d_{j2}} \log \left\{ \exp\left(\frac{\|d_j - d_j\|_1}{\sigma(\epsilon)}\right) \right\} 
= \frac{\Delta D_j}{\sigma(\epsilon)} 
\]

(4.20)

---

3Dwork et al. in [39] defined Eq. 4.20 as *Indistinguishability*, and here we name it as *Self Indistinguishability* in order to compare with the *Dependent Indistinguishability* of $D_j$. 

100
Self Indistinguishability represents the maximal difference of \( D_j \) caused by the modification of \( D_j \) itself. We further define the Dependent Indistinguishability of \( D_j \) induced by \( D_i \) as

\[
\text{Dependent Indistinguishability} = \max_{d_i} \log \left\{ \sum_{d_j} P(D_j = d_j | D_i = d_i) \exp \left( \frac{\|d_j - d_j^*\|_1}{\sigma(\epsilon)} \right) \right\} \tag{4.21}
\]

Dependent Indistinguishability evaluates the maximal expected difference in \( D_j \) caused by the modification of \( D_i \). Therefore,

\[
\rho_{ij} = \frac{\text{Dependent Indistinguishability}}{\text{Self Indistinguishability}} \tag{4.22}
\]

or in other words, \( \rho_{ij} \) evaluates the ratio of dependent indistinguishability of \( D_j \) induced by \( D_i \) and the self indistinguishability of \( D_j \).

To generalize our dependent perturbation mechanism, we consider an arbitrary query function \( Q \) and compute the dependent sensitivity of \( Q \) over \( D_j \) induced by the modification of \( D_i \) as

\[
DS^{Q}_{ij} = \rho_{ij} \Delta Q_j \tag{4.23}
\]

where \( \Delta Q_j \) is the sensitivity of \( Q \) with respect to the modification of \( D_j \) itself, i.e.,

\[
\Delta Q_j = \max_{d_{j1}, d_{j2}} \|Q(\cdots, d_{j1}, \cdots) - Q(\cdots, d_{j2}, \cdots)\|_1.
\]

Proof. As \( \rho_{ij} \) evaluates the extent of dependence between \( D_i \) and \( D_j \), the modification of \( D_i \) would imply modification of \( D_j \) as \( \rho_{ij} \Delta D_j \). Therefore, for any query function \( Q \), we have the corresponding sensitivity for \( D_j \) as \( \rho_{ij} \Delta Q_j \). Furthermore, we can prove

\[
DS^{Q}_i = \max_{d_{i1}, d_{i2}} \|Q([D_1, \cdots, d_{i1}, \cdots]) - Q([D_1, \cdots, d_{i2}, \cdots])\|_1 = \max_{d_{i1}, d_{i2}} \int_{d_{i1}}^{d_{i2}} \frac{\partial Q(D)}{\partial D_i} dD_i + \int_{d_{j1}}^{d_{j2}} \frac{\partial Q(D)}{\partial D_j} dD_j + \cdots \leq \Delta Q_i + \Delta Q_j + \cdots \leq \Delta Q_i +
\]
\[ \rho_{ij} \Delta Q_j = \sum_{j=c_{i1}}^{C_{iL}} \rho_{ij} \Delta Q_j. \] Therefore, the global sensitivity for publishing any query function \( Q \) on a dependent dataset is \( DS^Q = \max_i DS^Q_i = \sum_j \rho_{ij} \Delta Q_j. \)

Furthermore, we can generalize the dependent sensitivity to multiple users as

\[ DS^Q_i = \sum_{j=c_{i1}}^{C_{iL}} \rho_{ij} \Delta Q_j \] (4.24)

where \( C_{i1}, \ldots, C_{iL} \) represent the \( L \) tuples that are dependent with \( i \)-th tuple and \( \rho_{ii} = 1 \). \( DS^Q_i \) measures the dependent sensitivity of \( Q \) over all tuples in \( D \) caused by the modification of one individual tuple \( D_i \). We further derive the dependent sensitivity for the whole dataset as

**Theorem 14.** The dependent sensitivity for publishing any query \( Q \) over a dependent (correlated) dataset is

\[ DS^Q = \max_i DS^Q_i \] (4.25)

Finally, the dependent perturbation mechanism (DPM) for achieving \( \epsilon \)-dependent differential privacy is formalized as

**Theorem 15.** For any query function \( Q \) over an arbitrary domain \( D \) with dependent tuples, the mechanism \( \mathcal{A} \)

\[ \mathcal{A}(D) = Q(D) + \text{Lap}(DS^Q / \epsilon) \] (4.26)

gives \( \epsilon \)-dependent differential privacy.
4.5.3 Utility and Privacy Guarantees

While the privacy guarantees of $\epsilon$-differential privacy are well understood, the resulting utility due to the privacy mechanisms is often on a best-effort basis. In the following, we analyze the utility provided by our DPM. To do so, we consider a well known utility definition suggested by Blum et al. in [13].

Definition 10. $((\alpha, \beta)$-Accuracy): A randomization algorithm $A$ satisfies $(\alpha, \beta)$ accuracy for a query function $Q$, if $\max_D |A(D) - Q(D)| < \alpha$ with probability $1 - \beta$.

Based on the definition of $(\alpha, \beta)$-Accuracy, we have the utility guarantee for DPM as

Theorem 16. A DPM $A$ that satisfies $\epsilon$-dependent differential privacy would achieve $\max_D |A(D) - Q(D)| < \alpha$ with probability $1 - \exp(-\frac{\epsilon \alpha}{DSQ})$.

Proof.

\[
P(\max_D |A(D) - Q(D)| > \alpha) \leq \beta \implies P(\max_D |\text{Lap}(\frac{DSQ}{\epsilon})| > \alpha) \leq \beta
\]

\[
\implies P(\text{Lap}(\frac{DSQ}{\epsilon}) > \alpha) + P(\text{Lap}(\frac{DSQ}{\epsilon}) < -\alpha) \leq \beta
\]

\[
\implies 2 \int_0^\alpha t \exp(-\frac{\epsilon t}{DSQ}) dt \leq \beta
\]

\[
\implies \exp(-\frac{\epsilon \alpha}{DSQ}) \leq \beta
\]

(4.27)

Furthermore, we theoretically demonstrate the utility and privacy superiority of DPM over the baseline approach in Section 4.5.1.

Lemma 1. Under the same privacy budget $\epsilon$, DPM achieves better utility performance than the baseline approach.
Proof. Given $\epsilon_{DPM} = \epsilon_{base}$, we have $\beta_{DPM} = 1 - \exp(-\frac{\epsilon_0}{DS^Q}) > 1 - \exp(-\frac{\epsilon_0}{L\Delta Q}) = \beta_{base}$ (since $DS^Q = \max_i \sum_j \rho_{ij}\Delta Q_j \leq L\Delta Q$). Therefore, DPM achieves smaller query errors and thus better utility performance.

**Lemma 2.** Under the same $(\alpha, \beta)$-accuracy, DPM achieves better privacy performance than the baseline approach.

Proof. Given $\beta_{DPM} = \beta_{base}$, we have $\epsilon_{DPM} = -\frac{DS^Q \log(\beta)}{\alpha} < -\frac{L \log(\beta)}{\alpha} = \epsilon_{base}$ (since $DS^Q = \max_i \sum_j \rho_{ij}\Delta Q_j \leq L\Delta Q$). Therefore, our DPM results in better privacy performance.

4.5.4 Implications of Dependence Coefficient in System Design

We now discuss a practical challenge regarding the computation of the dependence coefficient. The dependence coefficient $\rho_{ij}$ between two tuples $D_i, D_j$ relies on the probabilistic models of the statistical data. Thus, it is difficult to compute $\rho_{ij}$ reliably unless the probabilistic models are known. Here, we provide several effective strategies to compute $\rho_{ij}$, as guidelines for a data publisher to select a proper privacy model for her own setting.

4.5.4.1 Complete Knowledge of Dependence Relationship

The first type of analysis assumes that the data publisher has access to the complete knowledge of the dependence relationship between tuples in advance. Sen et al. [108] computed the dependence relationship among tuples using an appropriately constructed probabilistic graphical model. Their method relies on a fully known probabilistic database in which the dependent tuples have associated probabilities. Using
the dependent probabilities, we can compute the dependence coefficient according to Eq. 4.19.

4.5.4.2 Knowledge About Data Generation

However, the entire dependence information between tuples is not always available to the data publisher. Under certain scenarios where the data generation process is known, the data publisher can estimate the dependence relationship by carefully analyzing the data generating process. For example, in [71], assuming that the social network generation model is known, extensive experiments and analysis were described to estimate the dependence relationship of the tuples.

Even in the absence of direct dependence information, analysis can still be carried out to estimate an upper bound on the dependence coefficient based on auxiliary information regarding the data (e.g., by using the Gowalla social datasets in Section 4.3).

Here, we consider to utilize the friend-based model in [8] to compute the probabilistic dependence relationship, where a user’s location can be estimated by her friend’s location based on the distance between their locations. Specifically, the probability of a user \( j \) locating at \( d_j \) when her friend \( i \) is locating at \( d_i \) is

\[
P(D_j = d_j | D_i = d_i) = a(\|d_j - d_i\|_1 + b)^{-c}
\]  

(4.28)

where \( a > 0, b > 0, c > 0 \). The effectiveness of the probabilistic dependence relationship in Eq. 4.28 will be verified on multiple real-world datasets in Section 4.6. We believe that alternate potential strategies for dependence relationship analysis will be an impactful direction for future work. But no matter which method is applied, our
goal is to evaluate fine-grained probabilistic dependence relationship among tuples for designing data sharing algorithms that satisfy $\epsilon$-dependent differential privacy.

### 4.5.4.3 Challenges in Realistic Scenario

Furthermore, we carefully analyze the influence of inaccurate computation in $\rho_{ij}$ on the overall performance of our DPM. We believe that designers are well-placed to compute $\rho_{ij}$. If $\rho_{ij}$ is overestimated, DPM is conservative and continues to provide rigorous DDP privacy guarantees. In case of underestimation of $\rho_{ij}$, there are two cases. In the first case, if our estimated $\rho'_{ij}$ is larger than the expectation of the adversary who has access to certain auxiliary information, our DPM can still continue to provide rigorous DDP guarantees. However, in second case, in which the underestimation of $\rho_{ij}$ is smaller than the adversary’s expectation, we may not achieve the DDP guarantees, but would still provide better privacy than the traditional DP mechanism. Furthermore, we consider a natural relaxation of dependent differential privacy to incorporate such imperfect estimation of $\rho_{ij}$.

**Definition 11.** (($\epsilon, \delta$)-Dependent Differential Privacy) A randomized algorithm $A$ provides ($\epsilon, \delta$)-dependent differential privacy, if for any pair of dependent neighboring databases $D(L, \mathcal{R})$ and $D'(L, \mathcal{R})$ and any possible output $S$, we have

\[
P(A(D(L, \mathcal{R})) = S) \leq \exp(\epsilon) P(A(D'(L, \mathcal{R})) = S) + \delta
\]

where $D(L, \mathcal{R}), D'(L, \mathcal{R})$ are dependent neighboring databases (recall Definition 8), based on the dependence size $L$ and their probabilistic dependence relationship $\mathcal{R}$.

Better accuracy (a smaller magnitude of added noise owing to the underestimation of $\rho_{ij}$) and generally more flexibility can often be achieved by relaxing the definition
of DDP in Eq. 9. Exploring such relaxations of DDP would be an interesting direction for future work.

## 4.6 Experimental Results

This section evaluates the performance of our proposed dependent data release algorithm on multiple real-world datasets (including Gowalla data in Section 4.3.1, the adult data in UCI Machine Learning Repository, and the large-scale Google+ data [50]). Our objectives are: 1) to show the privacy and utility superiority of our DPM over the state-of-the-art approaches, 2) to study the impact of enforcing DDP on the data in terms of machine learning queries and graph queries, and 3) to analyze the resistance of DPM to inference attacks described in Section 4.3.3.

### 4.6.1 Privacy and Utility Guarantees

Consider the application scenario in Fig. 4.3.2, where the data provider publishes the perturbed $K$-means centroids of the Gowalla location dataset while preserving the privacy of each individual data. Since the Gowalla dataset contains no associated probabilistic distributions or data generating process, we use the general dependent model in Eq. 4.28 to compute the dependence coefficient $\rho_{ij}$ in Eq. 4.19 by setting $a = 0.0019, b = 0.196, c = 1.05$ as empirically determined according to [8]. Then, the global sensitivity $DSQ$ can be computed according to Eq. 4.24 and Eq. 4.25.

Fig. 4.7(a) analyzes the $(\alpha, \beta)$-accuracy in Definition 10 under various privacy-preserving level $\epsilon$. We can see that under the same $\alpha$ and $\epsilon$, our DPM has much lower $\beta$ than the baseline approach (where the dependence size $L$ is set to be equal to the number of tuples) and the approach of Zhu et al. in [139], i.e., $\|A(D) - Q(D)\|_1 \leq \alpha$. 107
Figure 4.7. Comparison of (a) utility and (b) privacy performance of different perturbation mechanisms.

with higher probability $1 - \beta$. Note that the approach of Zhu et al. [139] utilized the linear relationships among tuples for correlated data publishing, which does not satisfy any rigorous privacy metric. Therefore, DPM achieves much better accuracy than the existing approaches, and such advantage increases with a larger privacy preserving level $\epsilon$. When $\alpha = 1000, \epsilon = 1$, the probability of $\|A(D) - Q(D)\|_1 < \alpha$ for DPM reaches nearly 1 while in comparison this probability is approximately 0 for the other methods. Similarly, Fig. 4.7(b) demonstrates that DPM also provides significantly better privacy performance than the existing approaches under the same
utility constraint. Therefore, DPM shows significant privacy and utility superiority over the state-of-the-art approaches as theoretically analyzed in Lemmas 1, 2.

4.6.2 Application Quality of Service

4.6.2.1 Clustering

In addition to the $(\alpha, \beta)$-accuracy, we further evaluate the utility performance of DPM by sharing the perturbed query results with real-world applications that use machine learning algorithms and analyzing the quality of service for these applications.

![Figure 4.8. Clustering accuracy under different perturbation methods.](image)

We evaluate the clustering accuracy of the dependent differentially private K-means centroids based on the cross-validation mechanism. We randomly select $4/5$ data from the Gowalla location dataset for training the dependent differentially private centroids $\tilde{\mu} = [\tilde{\mu}_1, \cdots, \tilde{\mu}_k]$ and then apply the perturbed centroids to cluster the remaining $1/5$ location data. We repeat the cross-validation process for 1000 times and compare the average clustering performance of DPM with the state-of-the-art approaches. For a more comprehensive investigation, we evaluate the clustering accuracy of DPM under various privacy budget $\epsilon$.

From Fig. 4.8, we observe that DPM has significantly better clustering accuracy and
over the baseline approach and that proposed by Zhu et al. in [139]. The reason is that our DPM adds less noise to the K-means centroids by incorporating finer-grained dependence relationship among tuples. Therefore, for dependent datasets, DPM outperforms the state-of-the-art approaches in preserving the quality of service for real-world applications.

For $\epsilon = 0.9$, which corresponds to a fairly strong privacy guarantee, DPM achieves an acceptable clustering performance with nearly 80% accuracy, which is more than twice that of the other approaches. This indicates that DPM is capable of retaining the application quality of service while satisfying a suitable privacy preserving requirement.

![Classification accuracy under different perturbation methods.](image)

4.6.2.2 Classification

We also apply our DPM to the widely used classification query in machine learning, by designing the dependent differentially private support vector machine (SVM) to the Adult dataset in UCI Machine Learning Repository[^1]. Detailed process for applying DP to SVM classification can be found in [21]. This dataset contains multiple users’ profiles and are labeled according to the users’ salaries. By deleting those records

[^1]: https://archive.ics.uci.edu/ml/datasets/Adult/
with missing attributes, we extract a new dataset with 30,269 tuples and each tuple has 14 attributes.

To compute the dependence coefficient, we first construct an affinity graph based on the similarities between the users’ profiles, where an edge exists for a pair of users $i$ and $j$ if $\frac{||d_i^T d_j||_1}{||d_i||_1 ||d_j||_1} > 0.8$ ($d_i, d_j$ are the profiles of tuple $i, j$ respectively). Similarly, we compute the dependence coefficient $\rho_{ij}$ for users $i$ and $j$ according to Eqs. 4.28, 4.19, and 4.25. Fig. 4.9 shows that our DPM has much better classification accuracy than the other methods by considering fine-grained dependence relationship. For $\epsilon = 0.9$, which represents a strong privacy level, DPM achieves an accurate classification performance with 85% accuracy, which is more than twice that of the other approaches. Therefore, DPM could provide an acceptable application quality of service while providing rigorous privacy guarantees.

![Degree distribution accuracy under different perturbation methods.](image)

4.6.2.3 Degree Distribution

We further consider a graph query whose result is to publish the degree distribution of a large-scale Google+ dataset [50]. The Google+ dataset is crawled from July 2011 to October 2011, which consists of 28,942,911 users and 947,776,172 edges and thus contains a broad degree distribution. The degree distribution of a graph is a histogram.
partitioning the nodes in the graph by their degrees \cite{119}, and it is often used to describe the underlying structure of social networks for the purposes of developing graph models and making similarity comparisons between graphs. Detailed process for applying differential privacy on degree distribution can be found in \cite{119}. In addition to the social graph, an auxiliary data is also provided in this dataset with users’ attributes such as Employment and Education. To compute the dependence coefficient, we construct an affinity graph based on the similarities between the users’ profiles, where an edge is added for a pair of users $i$ and $j$ if $\|d_i^T d_j\|_1 > 0.8$ and $d_i, d_j$ represent the profiles of tuple $i, j$ respectively in the auxiliary data. Similarly, we compute the dependence coefficient $\rho_{ij}$ for users $i$ and $j$ according to Eqs. 4.28, 4.19, 4.25. Denoting $C(D)$ and $C'(D)$ as the true degree distribution and the perturbed degree distribution respectively, we define the accuracy for publishing $C'(D)$ as $1 - \frac{\|C(D) - C'(D)\|_1}{\|C(D) + C'(D)\|_1}$. By considering fine-grained dependence relationship, our DPM has significantly higher accuracy for publishing dependent differentially private degree distribution of the social graph than the other methods, with almost 10x improvement as shown in Fig. 4.10.

4.6.3 Summary for the Experimental Analysis

- DPM provides significant privacy and utility gains compared to the state-of-the-art approaches. Therefore, we can select a suitable privacy budget $\epsilon$ to achieve an optimal privacy and utility balance for DPM.

- DPM is more than 2x accurate in computing the K-means clustering centroids and the SVM classifier, and more than 10x accurate in publishing degree distribution of large-scale social network, compared with existing approaches (which may not even provide rigorous privacy guarantees). These results demonstrate the effectiveness of DPM in real-world query answering for network data.
4.7 Chapter Summary

Differential privacy provides a formal basis for expressing and quantifying privacy goals. For these reasons there is an emerging consensus in the privacy community around its use and various extensions are being proposed. However, existing DP metric/mechanism does not explicitly model data dependence (correlation) existing in the database. In this work, we show that social networks that exist between users can be used to extract more sensitive location information from noisy query results generated by DP mechanisms than expected. To defend against such attacks, we introduced a generalized dependent differential privacy framework that incorporates probabilistic dependence relationship between data and provides rigorous privacy guarantees. We further propose a dependent perturbation mechanism and rigorously prove that it can achieve the privacy guarantees. Our evaluations over multiple large-scale real datasets and multiple query classes show that the dependent perturbation scheme performs significantly better than state-of-the-art approaches used for providing differential privacy.
Chapter 5

Investigating Statistical Privacy Frameworks from the Perspective of Hypothesis Testing

The field of statistical data privacy aims at sanitizing the data before its release so that individual privacy can be protected while enabling data analysis. Over the last decade, differential privacy (DP) has emerged as the gold standard of a rigorous and provable privacy framework. However, there are very few practical guidelines on how to apply differential privacy in practice, and a key challenge is how to set an appropriate value for the privacy parameter $\epsilon$. In this work, we employ a statistical tool called hypothesis testing for discovering useful and interpretable guidelines for the state-of-the-art privacy-preserving frameworks. We formalize and implement hypothesis testing in terms of an adversary’s capability to infer mutually exclusive sensitive information about the input data (such as whether an individual has participated or not) from the output of the privacy-preserving mechanism. We quantify the success of the hypothesis testing using the precision-recall-relation, which provides an
interpretable and natural guideline for practitioners and researchers on selecting $\epsilon$. Our key results include a quantitative analysis of how hypothesis testing can guide the choice of the privacy parameter $\epsilon$ in an interpretable manner for a differentially private mechanism and its variants. Importantly, our findings show that an adversary’s auxiliary information – in the form of prior distribution of the database and correlation across records and time – indeed influences the proper choice of $\epsilon$. Finally, we also show how the perspective of hypothesis testing can provide useful insights on the relationships among a broad range of privacy frameworks including differential privacy, Pufferfish privacy, Blowfish privacy, dependent differential privacy, inferential privacy, membership privacy and mutual-information based differential privacy.

### 5.1 Motivation

An important thread of research in the security community has investigated approaches for protecting the privacy of sensitive user data while enabling data analytics [26, 34, 49, 58, 72, 81, 83]. Among these approaches, differential privacy (DP) [18, 34, 37, 39, 41, 43, 69, 88, 91, 93] has emerged as the gold standard for providing rigorous and provable privacy protection for individuals. A differentially-private mechanism guarantees that the participation of any individual in the database does not significantly change the output of the mechanism, where the degree of change is quantified by a tunable privacy parameter $\epsilon$.

While the concept of differential privacy has received considerable attention in the last decade, including industry and government adoption (e.g., Google, Apple, and US Census), there are very few guidelines on how to apply it in practice [52, 118]. As illustrated by the recent controversy surrounding Apple’s implementation of differential privacy [118], a key challenge facing system designers and researchers
is how to set an appropriate value of $\epsilon$. Dwork and Smith have also identified this as an open research direction [43]. Specifically, they considered the choice of $\epsilon$ as essentially a social question. However, existing tools provide only a limited support for understanding this social question. In addition, it has been observed in [23, 58, 71, 72, 83, 139] that the appropriate choice of $\epsilon$ may also be affected by the existence of auxiliary information. To address these challenges, we aim to provide a rigorous and quantitative procedure to investigate the choice of an appropriate value of $\epsilon$, from the perspective of adversaries’ hypothesis testing. In our work, we also consider adversaries that have access to arbitrary auxiliary information, especially focusing on their influence on the choice of $\epsilon$.

**Contributions.** In order to convincingly determine an appropriate value of $\epsilon$ and analyze the effect of auxiliary information on this choice, we need an interpretable notion of how much information is leaked about individuals from the mechanism outputs. In other words, we need a tool that can relate the value of $\epsilon$ to a more semantically meaningful and, crucially, measurable quantity. Only a limited number of previous works [61, 74, 76] have investigated the question of how to select a proper value of $\epsilon$, but these approaches either require complicated economic models or lack the analysis of adversaries with arbitrary auxiliary information (see Section 2.3.3 for more details). Our work is inspired by the interpretation of differential privacy via hypothesis testing, initially introduced by Wasserman and Zhou [53, 67, 128]. However, this interpretation has not been systematically investigated before in the context of our research objective, i.e., reasoning about the choice of the privacy parameter $\epsilon$ (see Section 2.3.2 for more details).

We consider hypothesis testing [5, 99, 129] as the tool used by the adversary to infer sensitive information of an individual record (e.g., the presence or absence of a record in the database for unbounded DP) from the outputs of privacy mechanisms.
In particular, we employ the precision-recall (PR) relation from the perspective of hypothesis testing by the adversary as the measurable quantity of interest. The PR relation considers the trade-off between the precision (the fraction of examples classified as input records that are truly existing in the input data for unbounded DP) and the recall (the fraction of truly existing input records that are correctly detected for unbounded DP) from the adversary’s perspective.

With this context of hypothesis testing, we consider three research questions in this work: how do we set the value of $\epsilon$, how does auxiliary information affect the choice of $\epsilon$, and can hypothesis testing be used to systematically compare across heterogeneous privacy frameworks? We introduce our concrete approach to address these questions below.

**Investigating Differential Privacy.** To explore the choice of an appropriate value of $\epsilon$, we consider an adversary who tries to infer the existence of a record $d_i$ in the database $D$ from the output of a differentially private mechanism $A(D)$. Our threat model is an adversary who uses hypothesis testing with the Neyman-Pearson criterion [102], which is one of the most powerful criteria in hypothesis testing, on the noisy query results obtained by DP mechanisms. We focus on using the Neyman-Pearson criterion for the Laplace perturbation mechanism [39] in order to perform a concrete analysis. We also show how to generalize our approach to other mechanisms such as the Gaussian perturbation mechanism. Particularly, we leverage the PR-relation and the corresponding $F_\beta$-score (the weighted harmonic average of precision and recall [123]) as effective metrics to quantify the performance of adversaries’ hypothesis testing, which can provide a natural and interpretable guideline for selecting proper privacy parameters by system designers and researchers. Furthermore, we extend our analysis on unbounded DP to bounded DP and the approximate $(\epsilon, \delta)$-DP.
Impact of Auxiliary Information. The conjecture that auxiliary information can influence the design of DP mechanisms has been made in prior work \cite{23,58,71,72,83,139}. We therefore investigate the adversary’s capability based on hypothesis testing under three types of auxiliary information: the prior distribution of the input record, the correlation across records, and the correlation across time. Our analysis demonstrates that the auxiliary information indeed influences the appropriate selection of $\epsilon$. The results suggest that, when possible and available, the practitioners of DP should explicitly incorporate adversary’s auxiliary information into the parameter design of their privacy frameworks. Hence, our results provide a rigorous and systematic answer to the important question posted by Dwork and Smith \cite{43}.

Comparison of Statistical Privacy Frameworks. In addition to the two primary questions regarding differential privacy, we also extend our hypothesis testing analysis to a comparative study of a range of state-of-the-art privacy-preserving frameworks \cite{26,49,58,72,81,83}. Some of these frameworks \cite{23,58,71,72,139} have considered adversaries with auxiliary knowledge in their definitions, but no prior work has applied a common technique to compare and understand their relationship among each other and with differential privacy.

Overall, our work makes the following contributions.

- We investigate differential privacy from the perspective of hypothesis testing by the adversary who observes the differentially private outputs. We comprehensively analyze (i) the unbounded and (ii) bounded scenarios of DP, and (iii) $(\epsilon, \delta)$-DP.

- We theoretically derive the PR-relation and the corresponding $F_{\beta \text{score}}$ as criteria for selecting the value of $\epsilon$ that would limit (to the desired extent) the
adversary’s success in identifying a particular record, in an interpretable and quantitative manner.

- We analyze the effect of three types of auxiliary information, namely, the prior distribution of the input record, the correlation across records, and the correlation across time, on the appropriate choice of $\epsilon$ via the hypothesis testing by the adversary.

- Furthermore, we systematically compare the state-of-the-art statistical privacy notions from the perspective of the adversary’s hypothesis testing, including Pufferfish privacy [72], Blowfish privacy [58], dependent differential privacy [83], membership privacy [81], inferential privacy [49] and mutual-information based differential privacy [26].

The frequently used notations in this chapter are listed in Table 5.1.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D$</td>
<td>Random variable for the input database</td>
</tr>
<tr>
<td>$D, D'$</td>
<td>Two neighboring databases (instances of $D$)</td>
</tr>
<tr>
<td>$d_i$</td>
<td>The $i$-th record in the database</td>
</tr>
<tr>
<td>$d_{-i}$</td>
<td>All other records except for $d_i$ in the database</td>
</tr>
<tr>
<td>$Q(\cdot)$</td>
<td>Query function</td>
</tr>
<tr>
<td>$A(\cdot)$</td>
<td>Privacy-preserving mechanism</td>
</tr>
<tr>
<td>$o$</td>
<td>Noisy output</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>Privacy parameter</td>
</tr>
<tr>
<td>$h_0/h_1$</td>
<td>Hypothesis corresponding to $D/D'$</td>
</tr>
<tr>
<td>$Aux$</td>
<td>Auxiliary information of the adversary</td>
</tr>
<tr>
<td>$\theta$</td>
<td>Threshold for the noisy output</td>
</tr>
<tr>
<td>$\Lambda(\cdot)$</td>
<td>Likelihood ratio</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Threshold for the likelihood ratio</td>
</tr>
<tr>
<td>$[t]$</td>
<td>Timestamps from 1 to $t$</td>
</tr>
<tr>
<td>$\rho_p$</td>
<td>Coefficient for prior distribution of the input data</td>
</tr>
<tr>
<td>$\rho_c$</td>
<td>Coefficient for correlation across records</td>
</tr>
<tr>
<td>$\rho_t$</td>
<td>Coefficient for correlation across time</td>
</tr>
</tbody>
</table>
5.2 Methodology Overview

In this section, we describe our quantitative procedure to investigate the choice of an appropriate value of $\epsilon$ from the perspective of the adversary’s hypothesis testing (shown in Figure 5.1). Specifically, we discuss the threat model, the quantification method we utilize and the generality of our approach as follows.

**Threat Model:** In this work, we consider the standard adversary in DP that aims to infer the existence of a particular record (for unbounded scenario) or the true value of a record (for bounded scenario) from the noisy output of DP mechanisms. The adversary also has access to the values of all the other records of the input database and/or other auxiliary information such as the prior distribution of the input database and the correlation across records and time. We assume that the adversary exploits the Neyman-Pearson criterion in hypothesis testing to infer sensitive information of a particular record. Note that our analysis considers an information-theoretic/unbounded adversary (and not probabilistic polynomial time (PPT) adversary [122]).

Next, let us briefly describe how we apply the hypothesis testing in our setting. Assume that the adversary has access to the noisy result of DP mechanisms $A(D) = o$ and tries to infer the existence of a record $d_i$ in the database $D = [d_0, d_1, \cdots]$, where the neighboring database $D'$ assumes the non-existence of the record $d_i$ (for unbounded DP). For the bounded scenario, one database $D'$ can be obtained from its neighbor $D$ by replacing one record with a different value. We use a random variable $\mathcal{D}$ to represent the true database which is unknown to the adversary. The adversary would assume the following two hypotheses:

$$H = \begin{cases} h_0 : \mathcal{D} = D' \\ h_1 : \mathcal{D} = D \end{cases}$$

(5.1)
When the adversary observes the noisy result $A(D) = o$, he/she tries to distinguish the two events $D = D$ and $D = D'$ through analyzing the two posterior probabilities of $P(D = D|A(D) = o, d_{-i}, Aux)$ and $P(D = D'|A(D) = o, d_{-i}, Aux)$, where $d_{-i}$ represent the values of all the other records in the input database and $Aux$ represents the auxiliary information that may be accessible to the adversary (which will be discussed in details in Section 5.4). Using the Neyman-Pearson criterion (instead of directly measuring the statistical difference between the probabilities of neighboring databases), the adversary determines whether to accept $D$ or $D'$ (more details will be described in Sections 5.3, 5.4, 5.5) and we denote this detection result as $\hat{D}$.

**Leveraging PR-relation as Quantification Metric:** We utilize the precision-recall (PR)-relation as an effective metric to quantify the adversary’s capability of
hypothesis testing, which can also serve as a practical guideline for selecting proper values of the privacy parameter $\epsilon$. In statistics, *precision* denotes out of those predicted positive how many of them are actually positive, and *recall* denotes the fraction of the true positives that are labeled as positive. We leverage the PR-relation to quantify the adversary’s hypothesis testing (which has not been explored in previous works [53, 67, 128]) since it is more useful in practice for problems where the risk for the two hypotheses are different. In our setting, the more critical class corresponds to the truly existing records of the input database since the adversary’s detection of these records is more serious than that of non-existing records for unbounded DP. Specifically for our problem, *precision* and *recall* are defined as

\[
\text{precision} = P(h_1|\hat{D} = D) = P(D = D|\hat{D} = D) \\
\text{recall} = P(\hat{D} = D|h_1) = P(\hat{D} = D|D = D)
\]  

From Eq. 5.2, we know that *precision* is the probability that hypothesis $h_1$, which the adversary’s hypothesis testing says is true, is indeed true; and *recall* is the probability that hypothesis $h_1$, which is indeed true, is detected as true by the adversary’s hypothesis testing. Note that the randomness in the process of computing *precision* and *recall* comes from the statistical noise added in DP mechanisms. Therefore, our analysis aims to quantify the capability of the adversary that leverages the hypothesis testing technique in identifying any specific record from DP outputs.

PR-relation quantifies the actual detection accuracy of the adversary, which has a one-to-one correspondence with the false alarm rate $P_{FA}$ and the missed detection rate $1 - P_{TD}$ [67]. Different from Kairouz et al. [67] that quantifies the relative relationship between $P_{FA}$ and $1 - P_{TD}$, we obtain explicit expression of the precision and recall of the adversary that exploits the Neyman-Pearson criterion (in
Sections 5.3.1.3, 5.4.1, 5.4.2, 5.4.3). It is also interesting to note that there exists a one-to-one correspondence between PR-relation and the receiver operating characteristic (ROC) \cite{28}. Therefore, our analysis in the domain of PR-relation can be directly transferred to the domain of ROC.

**Theorem 17.** The Neyman-Pearson criterion characterizes the optimal adversary that can achieve the best PR-relation.

**Proof.** Achieve the Minimal $P_{FA}$ under a Given Level of $P_{TD}$ (recall): For a given false alarm rate $P_{FA}$, the maximal true detection rate $P_{TD}$ can be achieved according to the Neyman-Pearson criterion (Definition 3). Furthermore, note that the maximal $P_{TD}$ is not decreasing with the increasing of $P_{FA}$. That is to say, under a given level of the true detection rate $P_{TD} = P(\hat{D} = D|D = D) = P(\hat{D} = D, D = D)/P(D = D)$, the adversary implementing the Neyman-Pearson criterion can achieve the minimal false alarm rate $P_{FA} = P(\hat{D} = D|D = D') = P(\hat{D} = D, D = D)/P(D = D')$. Therefore, we obtain a fixed $P(\hat{D} = D, D = D)$ and a minimal $P(\hat{D} = D, D = D')$.

Achieve the Maximal Precision under a Given Level of $P_{TD}$ (recall): Since the recall and $P_{TD}$ are equivalent to each other, we know under any fixed level of recall ($P_{TD}$), the precision $P(D = D|\hat{D} = D) = P(\hat{D} = D, D = D)/P(\hat{D} = D, D = D) + P(\hat{D} = D, D = D')$ is maximized.

Therefore, the Neyman-Pearson criterion can achieve the maximal precision under any given level of recall, thus characterizing the optimal adversary that can achieve the best PR-relation.

In addition, we leverage $F_\beta$ score (weighted harmonic average of precision and recall), to further quantify the relationship between the adversary’s hypothesis testing and the
privacy parameter $\epsilon$, which also provides an interpretable guideline to practitioners and researchers for selecting $\epsilon$. $F_{\beta\text{score}} = \frac{1}{(1+\beta^2)\text{precision} + (1+\beta^2)\text{recall}}$ (for any real number $\beta > 0$) is an example for quantifying the PR-relation in order to understand the adversary’s power in a more convenient manner, since it combines the two metrics, precision and recall, into a single metric. However, this combination has the potential of information loss of the PR-relation which broadly covers the adversary’s hypothesis testing in the entire space (more discussions in Section 5.3.1.4). Note that every step of our analysis in Figure 5.1 is accurate in quantifying the adversary’s hypothesis testing under the Neyman-Pearson criterion using the metrics of PR-relation and $F_{\beta\text{score}}$.

**Generalizing to Other Privacy Mechanisms and Practical Adversaries:** We further generalize our analysis of the conventional $\epsilon$-DP to its variants such as $(\epsilon, \delta)$-DP (adopting the Gaussian perturbation mechanism), more advanced privacy notions, and also adversaries with auxiliary information. Our analysis shows that the adversary’s auxiliary information in the form of prior distribution of the input database, the correlation across records and time can affect the relationship between the two posterior probabilities (corresponding to the two hypotheses made by the adversary), thus impacting the proper value of privacy parameter $\epsilon$.

In this section, we theoretically analyze the capability of the adversary’s hypothesis testing for inferring sensitive information of a particular record from DP outputs. Specifically, we implement our analysis on the unbounded and bounded scenarios of DP and $(\epsilon, \delta)$-DP.
5.3 Quantification of DP from the Adversary’s Hypothesis Testing

5.3.1 Quantification of Unbounded DP

5.3.1.1 Hypothesis Testing Problem

Recall that the Neyman-Pearson criterion \(^{[102]}\) aims to maximize the true detection rate of the hypothesis test given a constrained false alarm rate (Definition \(^{[3]}\)). Following our threat model and methodology in Section \(^{[5.2]}\) the adversary would assume the following two hypotheses, corresponding to the presence or the absence of a record \(d_i\):

\[
H = \begin{cases} 
  h_0 : d_i \text{ does not exists in } \mathcal{D}, \text{ i.e., } \mathcal{D} = \mathcal{D}' \\
  h_1 : d_i \text{ exists in } \mathcal{D}, \text{ i.e., } \mathcal{D} = \mathcal{D} 
\end{cases}
\]  

(5.3)

This is clearly the unbounded DP setting (we will analyze the bounded DP setting and other DP variations in the next subsections.). After observing the noisy query result \(\mathcal{A}(\mathcal{D}) = o\), the adversary tries to distinguish the two events \(\mathcal{D} = \mathcal{D}\) and \(\mathcal{D} = \mathcal{D}'\) by analyzing the corresponding posterior probabilities of \(P(\mathcal{D} = \mathcal{D}|\mathcal{A}(\mathcal{D}) = o, d_{-i})\) and \(P(\mathcal{D} = \mathcal{D}'|\mathcal{A}(\mathcal{D}) = o, d_{-i})\). Following the Bayes’ rule of

\[
P(\mathcal{D} = \mathcal{D}|\mathcal{A}(\mathcal{D}) = o, d_{-i}) = \frac{P(\mathcal{A}(\mathcal{D}) = o|\mathcal{A}(\mathcal{D}) = o, \mathcal{D} = \mathcal{D})P(\mathcal{D} = \mathcal{D})}{P(\mathcal{A}(\mathcal{D}) = o|d_{-i})} = \frac{P(\mathcal{A}(\mathcal{D}) = o)P(\mathcal{D} = \mathcal{D})}{P(\mathcal{A}(\mathcal{D}) = o|d_{-i})},
\]

(5.4)

we know that distinguishing the two posterior probabilities is equivalent to differentiating the two conditional probabilities of \(P(\mathcal{A}(\mathcal{D}) = o)\) and \(P(\mathcal{A}(\mathcal{D}') = o)\)\(^{[1]}\) for

\(^{1}P(\mathcal{A}(\mathcal{D}) = o)\) represents the same conditional probability as \(P(\mathcal{A}(\mathcal{D}) = o|\mathcal{D} = \mathcal{D})\).
adversaries that have no access to the prior distribution of the input database and thus assume a uniform prior, i.e., $P(D = D) = P(D = D')$.

5.3.1.2 Decision Rule

Adversary’s Hypothesis Testing for Scalar Query: We first consider the situation where the query output is a scalar and then generalize our analysis to vector output. It is worth noting that DP is defined in terms of probability measures but likelihood is defined in terms of probability densities. However, we use them interchangeably in this chapter (we can change the probability densities to the probability measures through quantizing over the query output for instance). Without loss of generality, we assume $Q(D) \geq Q(D')$. Then, we have the following theorem.

Theorem 18. Applying Neyman-Pearson criterion of maximizing the true detection rate under a given requirement of false alarm rate $\alpha$ in Definition 3 is equivalent to the following hypothesis testing which is of a simpler formulation: setting a threshold

$$\theta = \begin{cases} 
- \frac{\Delta Q \log 2\alpha}{\epsilon} + Q(D'), & \alpha \in [0, 0.5] \\
\frac{\Delta Q \log 2(1 - \alpha)}{\epsilon} + Q(D'), & \alpha \in (0.5, 1]
\end{cases}$$

(5.5)

for the output of LPM-based DP mechanisms $A(D) = o$, then the decision rule of the adversary’s hypothesis testing is $o_b \begin{cases} h_1 & \text{if } Q(D) \leq Q(D') \\
h_0 & \text{if } Q(D) > Q(D') \end{cases}$

Proof. Following the Neyman-Pearson Lemma [101], we utilize the likelihood ratio test [107, 125] to realize the Neyman-Pearson criterion. Therefore, we first compute the likelihood ratio $\Lambda(o)$ of the two hypotheses and set a threshold $\lambda$ on this ratio for the adversary’s decision. Under a given requirement of the false alarm rate $\alpha$, we

\footnote{The decision rule is $o_b \begin{cases} h_1 & \text{if } Q(D) \leq Q(D') \\
h_0 & \text{if } Q(D) > Q(D') \end{cases}$.
can uniquely determine the threshold $\theta$ of the noisy output. Then, we can compute $\lambda$ from $\theta$. Next, let us discuss each step of this proof in detail.

**Construct Likelihood Ratio Test:** Given the noisy scalar output $o = A(D) = Q(D) + Lap(\zeta) = Q(D) + Lap(\Delta Q/\epsilon)$ from the LPM, we can compute the likelihood ratio $\Lambda(o)$ corresponding to the two hypotheses defined in Eq. 5.3 as

$$
\Lambda(o) = \frac{P(A(D) = o)}{P(A(D') = o)} = \frac{\epsilon^{-\frac{\epsilon\|o - Q(D')\|}{\Delta Q}}}{\epsilon^{-\frac{\epsilon\|o - Q(D)\|}{\Delta Q}}} = \begin{cases} 
\exp(\epsilon) & \text{if } o > Q(D) \\
\exp\left(\frac{2\epsilon - Q'(D) - Q(D')}{\Delta Q}\right) & \text{if } o \in [Q(D'), Q(D)] \\
\exp(-\epsilon) & \text{if } o < Q(D') 
\end{cases} \tag{5.6}
$$

Assume the decision threshold for the likelihood ratio is $\lambda$, then the corresponding decision rule in Neyman-Pearson criterion is $\Lambda(o) \geq \lambda$.

**Uniquely Determine $\theta$ from $\alpha$:** Under a threshold $\theta$ on the noisy output $o$, $P_{FA}$ can be computed as $1 - \int_{\theta}^{\infty} P(A(D') = o)do = 1 - \int_{\theta}^{\infty} \frac{\epsilon e^{-\frac{\epsilon\|o - Q(D')\|}{\Delta Q}}}{2\Delta Q} dx$, which is $1 - \frac{1}{2} e^{-\frac{(\theta - Q'(D'))\epsilon}{\Delta Q}}$ if $\theta < Q(D')$, or $\frac{1}{2} e^{-\frac{(\theta - Q'(D'))\epsilon}{\Delta Q}}$ if $\theta \geq Q(D')$. Therefore, for a given requirement of the false alarm rate $\alpha$, we can obtain Eq. 5.5.

**Compute $\lambda$ from $\theta$:** Based on Eq. 5.6 we know that $\exp(-\epsilon) \leq \Lambda(o) = \frac{P(A(D) = o)}{P(A(D') = o)} \leq \exp(\epsilon)$. Therefore, it is sufficient to choose $\lambda$ such that $\exp(-\epsilon) \leq \lambda \leq \exp(\epsilon)$. Then, the decision rule becomes $e^{\frac{2\epsilon - Q(D) - Q(D')}{\Delta Q}} \geq \lambda \implies \frac{h_1}{h_0} = \frac{\log \lambda \Delta Q}{2\epsilon} + \frac{Q(D) + Q(D')}{2}$. \(3\)

Therefore, the threshold $\lambda$ for the likelihood ratio $\Lambda(o)$ can be computed from the threshold $\theta$ for the noisy output $o$ as

$$
\lambda = e^{\frac{2\theta - Q(D) - Q(D')}{\Delta Q}} \tag{5.7}
$$

\(3\)We consider the natural base for logarithm in this chapter.
Based on the analysis above, we know that there exists a one-to-one correspondence between the false alarm rate $\alpha$ and the threshold of the noisy output $\theta$. This uniquely determined $\theta$ satisfies the existence and sufficient conditions for achieving Neyman-Pearson Lemma (see Theorem 3.2.1 in [79]). Therefore, the Neyman-Pearson criterion in our setting is equivalent to making a decision rule of $o \overset{h_1}{\gtrapprox} \theta$ by setting a threshold $\theta$ to the noisy output $o$, which is of a simpler formulation.

Note that Theorem 18 holds for any adversary that aims to distinguish the neighboring databases, including those who have access to other auxiliary information such as the prior distribution of the input data and correlation across records and time (see Section 5.4).

**Adversary’s Hypothesis Testing for Vector Query:** Our analysis for the scalar query can be readily generalized to the vector query output according to the following theorem.

**Theorem 19.** The best performance of hypothesis testing that the adversary can achieve on the output of $\epsilon$-DP mechanisms $Q : \mathcal{D} \rightarrow \mathbb{R}^q$ is the same as that on the output of $q\epsilon$-DP scalar mechanisms. 

**Proof.** Compared to the scalar query, the privacy property of the vector query $Q : \mathcal{D} \rightarrow \mathbb{R}^q$ would decrease by a factor of $q$ based on the sequential composition theorem of DP [48]. Therefore, the adversary’s capability for hypothesis testing is increased to that under the scalar query with a privacy parameter of $q\epsilon$.

Based on Theorem 19, we focus our analyses on the scalar query output in the subsequent discussion.
Figure 5.2. Adversary’s hypothesis testing of detecting a particular record under unbounded DP. By setting a threshold $\theta$, the adversary’s decision rule is $o \gtrless \theta$.

5.3.1.3 Evaluating Hypothesis Testing Performance

As stated in Section 5.2, we interpret the DP constraint (Definition 1) in the context of hypothesis testing in terms of the precision-recall (PR)-relation. Based on Theorem 18, we show the detailed process of hypothesis testing in Figure 5.2. The red curve and green curve corresponds to the conditional probabilities of the two hypotheses $h_1 : \mathcal{D} = D$ and $h_0 : \mathcal{D} = D'$, respectively. The decision rule for the adversary’s hypothesis testing is $o \gtrless \theta$. Next, we define two probabilities of $P(\hat{\mathcal{D}} = D|h_1) = \int_\theta^{+\infty} P(\mathcal{A}(D) = o)do$ and $P(\hat{\mathcal{D}} = D|h_0) = \int_{\theta}^{+\infty} P(\mathcal{A}(D') = o)do$, and these two probabilities are highlighted by $RSR$ (red shaded region), $GSR$ (green shaded region) in Figure 5.2. For the LPM, we can compute $RSR$ and $GSR$ as follows.

$$
RSR = \begin{cases} 
0.5e^{-\frac{(\theta - Q(D))_+}{\Delta Q}}, & \theta \in [Q(D), +\infty) \\
1 - 0.5e^{-\frac{(\theta - Q(D))_+}{\Delta Q}}, & \theta \in (-\infty, Q(D)] \end{cases}
$$

(5.8)

$$
GSR = \begin{cases} 
0.5e^{-\frac{(\theta - Q(D'))_+}{\Delta Q}}, & \theta \in [Q(D'), +\infty) \\
1 - 0.5e^{-\frac{(\theta - Q(D'))_+}{\Delta Q}}, & \theta \in (-\infty, Q(D')] \end{cases}
$$

(5.9)

Based on Eq. 5.2 and Figure 5.2, we can compute the precision and recall as
precision = \frac{RSR}{RSR + GSR}, \quad recall = RSR \quad (5.10)

Next, we plot the PR-relation for the adversary’s hypothesis testing under different values of $\epsilon$ in Figure 5.3(a). Under each $\epsilon$, the PR-relation is generated using precision and recall values at different $\theta$ (thus different values of $\alpha$ in Neyman-Pearson criterion). From Figure 5.3(a) we have the following observations: 1) we find that the adversary’s capability to infer the existence of a particular record decreases as more noise is added (corresponding to a smaller value of $\epsilon$); 2) when the privacy parameter $\epsilon$ is very small (e.g., $\epsilon = 0.01$), we have $\text{precision} \approx 0.5$, close to the worst hypothesis testing of the adversary (random guessing); 3) when the privacy parameter $\epsilon$ is very large (e.g., $\epsilon = 5$), we have very high $\text{precision}$ and $\text{recall}$, close to the best hypothesis testing of the adversary (nearly exact inference). Next, we prove that this analysis of PR-relation is applicable for any query function Q in the following theorem.

**Theorem 20.** The PR-relation of the adversary’s hypothesis testing on the outputs of LPM is independent of the query function $Q$.

**Proof.** For any query function $Q$, we define $\psi = \frac{\theta - Q(D')}{\Delta Q}$. The PR-relation shown in Figure 3(a) is directly generated by varying $\theta \in (-\infty, +\infty)$ which is independent of $Q$. Because $\theta$ can be any real number, $\psi$ can also be viewed as a free variable that can take any real number. Then, we have $\frac{\theta - Q(D)}{\Delta Q} = \frac{\theta - Q(D') - \Delta Q}{\Delta Q} = \psi - 1$. Substituting into Eqs. 5.8, 5.9, we know that $RSR = 0.5e^{-(\psi-1)\epsilon}$ if $\psi \in [1, +\infty)$, or $1 - 0.5e^{(\psi-1)\epsilon}$ if $\psi \in (-\infty, 1)$, and $GSR = 0.5e^{-\psi\epsilon}$ if $\psi \in [0, +\infty)$, or $1 - 0.5e^{\psi\epsilon}$ if $\psi \in (-\infty, 0)$. Given a RSR, $\psi$ can be computed and then a fixed GSR can be computed accordingly, which is independent of $Q$. Therefore, the PR-relation generated by precision and recall values (Eq. 5.10) at different values of $\psi \in (-\infty, +\infty)$ is independent of the query function $Q$ and is only determined by $\epsilon$. \qed
Figure 5.3. (a) PR-relation and (b) the highest $F_{\beta \text{score}}$ of detecting a particular record under unbounded DP.

5.3.1.4 Choosing Proper $\epsilon$

The privacy parameter $\epsilon$ can relatively measure the privacy guarantees of DP mechanisms. However, choosing appropriate values for $\epsilon$ is non-trivial since its impact on the privacy risks of the input data in practice are not well understood. Our method of choosing $\epsilon$ considers the capability of the adversary’s hypothesis testing to identify any particular record as being in the database. Specifically, we provide guidelines for selecting proper values of $\epsilon$ using PR-relation and $F_{\beta \text{score}}$, respectively.

**Guidelines under PR-relation:** Our analysis by leveraging PR-relation generally covers the adversary’s trade-offs between precision and recall in the entire space. Figure 5.3(a) demonstrates that, with sufficiently low privacy budget $\epsilon$, an adversary’s
ability to identify an individual record in the database is indeed limited. Under a given requirement of trade-offs between precision and recall of the adversary’s hypothesis testing, the privacy mechanism designer can refer to the PR-relation obtained in Eqs. 5.8, 5.9, 5.10 and Figure 5.3(a) to choose an appropriate privacy budget $\epsilon$.

**Guidelines under $F_{\beta \text{score}}$:** Besides the PR-relation, we also leverage the $F_{\beta \text{score}}$, which is a weighted harmonic average of precision and recall, as another appropriate metric for quantifying the effectiveness of the adversary’s hypothesis testing. Furthermore, we can theoretically derive the highest $F_{\beta \text{score}}$ (by selecting a proper threshold $\theta$) that the adversary can achieve for arbitrary real number $\beta > 0$ as

$$
F_{\beta \text{score}}^* = \max_{\theta} \frac{1}{(1+\beta^2)_{\text{precision}} + \beta^2 (1+\beta^2)_{\text{recall}}} \\
= \begin{cases} 
1 + \frac{\beta^2}{2 + \beta^2}, & \epsilon < \log(1 + \beta^2) \\
(1 + \beta^2)(\sqrt{1 + 4\beta^2 e^\epsilon} - 1) & \epsilon \geq \log(1 + \beta^2)
\end{cases}
$$

and the detailed proof is deferred to the Appendix.

Next, we show $F_{\beta \text{score}}^*$ with varying privacy parameter $\epsilon$ in Figure 5.3(b). Our results in Eq. 5.11 and Figure 5.3(b) are accurate quantification of the adversary’s hypothesis testing from the perspective of $F_{\beta \text{score}}$, from which we observe that the adversary’s capability of inferring the existence of an individual record is generally enhanced with an increasing value of $\epsilon$.

$F_{\beta \text{score}}$ can be interpreted as a summary statistic for the PR-relation, which provides a more convenient way of quantifying the relationship between the adversary’s hypothesis testing and the privacy parameter $\epsilon$. Under a desired bound of $F_{\beta \text{score}}^*$ that the adversary’s hypothesis testing can achieve, the privacy mechanism practitioners can refer to Eq. 5.11 and Figure 5.3(b) to choose an appropriate value of $\epsilon$. Further-
more, we provide numerical bounds of $\epsilon$ under different requirements of $F_{\beta \text{score}}$ for commonly-used weights of $\beta \in [0.5, 2]$ in Table 5.2, as an easier way to look up for privacy practitioners.

When the summary statistics of the precision and recall are used (as opposed to using the full PR-relation information), such as the use of the $F_{\beta \text{score}}$, there is potential for information loss, especially in the regime corresponding to smaller values of $\epsilon$ (Eq. 5.11 and Figure 5.3(b)). It is interesting to note that $F^*_{\beta \text{score}}$ keeps the same for $\epsilon < \log(1 + \beta^2)$, and then monotonically increases with $\epsilon$ for $\epsilon \geq \log(1 + \beta^2)$. This turning point $\epsilon = \log(1 + \beta^2)$ approaches 0 for smaller values of $\beta$, making $F^*_{\beta \text{score}}$ closer to be monotonically increasing with $\epsilon$ thus capturing the privacy benefits of smaller values of $\epsilon$ (as shown Figure 5.3(b)).

Finally, we emphasize that the alternative approach of using the entire PR-relation to guide the selection of $\epsilon$ does not suffer from the limitations discussed above, and also shows privacy benefits of using smaller $\epsilon$ (smaller precision for a given recall as shown in Figure 3(a)).

### 5.3.1.5 Plausible Deniability Property

There are multiple ways to interpret semantics of DP guarantees such as hypothesis testing \[53, 67, 128\] and plausible deniability (Page 9 in Dwork \[34\], Page 2 in Dwork)
and Smith [43], Definition 1 in Dwork, McSherry, Nissim and Smith [39], Section 2 in Kasiviswanathan and Smith [69], Section 4 in Li et al. [81]). The potential of randomness providing plausible deniability was first recognized by Warner [127]. Bindschaedler et al. provide a formal definition of plausible deniability for data synthesis, compared to which DP is a stronger privacy guarantee [12].

**Definition 12.** (Plausible Deniability) [12] For any database $D$ with $|D| > k$ ($|D|$ is the number of records in $D$), and any record $y$ generated by a perturbation mechanism $\mathcal{M}$ such that $y = \mathcal{M}(d_1)$ for $d_1 \in D$, we state that $y$ is releasable with $(k, \gamma)$-plausible deniability, if there exist at least $k - 1$ distinct records $d_2, \ldots, d_k \in D \setminus d_1$ such that $\gamma^{-1} \leq \frac{P(\mathcal{M}(d_i) = y)}{P(\mathcal{M}(d_j) = y)} \leq \gamma$ for any $i, j \in \{1, 2, \ldots, k\}$.

We interpret DP as hypothesis testing — how well an adversary in DP can infer the existence of an individual record (unbounded DP) or the exact value of a record (bounded DP) in binary hypothesis testing problem involving two neighboring databases (Dwork [34], Dwork, McSherry, Nissim and Smith [39], Kifer and Machanavajjhala [71]). Theorem 18 demonstrates that the adversary implements the likelihood ratio test $\Lambda(o) = \frac{P(A(D) = o)}{P(A(D') = o)}$ to satisfy the Neyman-Pearson criterion and the decision rule in Eq. 5.5 is equivalent to $\Lambda(o) \gtrless \lambda$. Combining Eq. 5.5 and Eq. 5.7, we know that $\lambda = \frac{e^{-\epsilon}}{4\alpha^2}$ if $\alpha \in [0, 0.5]$, or $\frac{e^{-\epsilon}}{4(1-\alpha)^2}$ if $\alpha \in (0.5, 1]$. According to Definition 12, the plausible deniability also quantifies the likelihood ratio between two data (although it only considers the scenario of privacy-preserving data synthesis [12]). Therefore, our analysis of using hypothesis testing to guide selection of proper privacy parameters in DP has implicitly incorporated the plausible deniability of any individual records in the database (controlled by the maximum false alarm rate $\alpha$ in the Neyman-Person criterion). Furthermore, $\alpha$ determines a trade-off between the false alarm rate $P_{FA}$ and the true detection rate $P_{TD}$ (Definition 3). Therefore, our analysis of using PR-relation (which has a one-to-one correspondence with $P_{FA}, P_{TD}$)
and $F_{\beta \text{score}}$ (summary statistics of precision and recall) generated by varying $\alpha$ quantifies the randomness and the plausible deniability [127][12] of any individual records in the database.

### 5.3.2 Quantification of Bounded DP

Bounded DP corresponds to the setting where the neighboring databases differ in one (record’s) value and their size is the same. For simplicity, we first discuss a scenario where the $i$-th record of the input data $d_i$ can only take binary values $d_{i1}, d_{i2}$. Note that the hypothesis testing by the adversary in the bounded scenario is different from the unbounded scenario in that the adversary is no longer aiming to distinguish the absence/presence of a record, but to estimate the true value of a record. Thus, the adversary’s hypothesis testing now becomes:

$$H = \begin{cases} h_0 : & d_i = d_{i1} \\ h_1 : & d_i = d_{i2} \end{cases}$$

Comparing Eq. 5.3 and Eq. 5.12, we know that the two hypotheses for unbounded and bounded cases are different. However, according to Theorem 20, their corresponding $PR$-relation (and $F_{\beta \text{score}}$) are the same. This means, the hypothesis testing implemented by the adversary for bounded DP with binary records is the same as that for unbounded DP.

Next, we consider a more general scenario where $d_i$ takes multiple values $d_{i1}, d_{i2}, \cdots, d_{ik}$. Without loss of generality, we assume $Q(d_{i1}) \leq Q(d_{i2}) \leq \cdots \leq Q(d_{ik})$. Therefore, the distance between any two query results computed over two different values of $d_i$ is smaller than the sensitivity of the query

$$\Delta Q = \max \|Q(d_{ik}) - Q(d_{i1})\|.$$  

Since the inserted noise for satisfying DP is calculated
based on $\Delta Q$ (i.e., $\text{Lap}(\frac{\Delta Q}{\epsilon})$), we know that the hypothesis testing achieved by the adversary in distinguishing any two values of $d_i$ is not worse than distinguishing $d_{i_1}$ and $d_{i_k}$. We thus conclude that the best hypothesis testing of the adversary for the bounded scenario is the same as that for the unbounded scenario.

5.3.3 Quantification of $(\epsilon, \delta)$-DP

Approximate DP, also named $(\epsilon, \delta)$-DP \cite{37} is defined as $P(A(D) \in S) \leq \exp(\epsilon)P(A(D') \in S) + \delta$ for any neighboring databases $D, D'$. One of the most popular mechanisms to achieve $(\epsilon, \delta)$-DP is the Gaussian perturbation mechanism, where a Gaussian noise with zero mean and standard variant $\sigma = \sqrt{2 \log(1.25/\delta)/\epsilon}$ is added to the query output \cite{37,40}.

Similar to Section 5.3.1, we first derive the mechanism for the adversary’s hypothesis testing that satisfies the Neyman-Pearson criterion based on the following theorem (detailed proof is deferred to the Appendix).

**Theorem 21.** Applying Neyman-Pearson criterion in Definition 3 is equivalent to the following hypothesis testing which is of a simpler formulation: setting a threshold $\theta = \Phi^{-1}(1 - \alpha)\sigma + Q(D')$ (where $\alpha$ is the maximum $P_{FA}$ and $\Phi(\cdot)$ is the cumulative distribution function of the standard normal distribution) for the output of the Gaussian perturbation mechanism $o$, the decision for the adversary’s hypothesis testing is $o \geq \theta$.

**Proof.** According to the Neyman-Pearson Lemma \cite{101}, the likelihood ratio test \cite{107,125} can be utilized to achieve the Neyman-Pearson criterion. For an adversary with access to the noisy scalar output $o = A(D) = Q(D) + \mathcal{N}(2\log(1.25/\delta)\Delta Q/\epsilon)$, we can compute the likelihood ratio corresponding to the two hypotheses defined in Eq. 5.3.
Figure 5.4. PR-relation of detecting a particular record from $(\epsilon, \delta)$-DP results with varying (a) $\epsilon$ and (b) $\delta$, respectively.

Then, we can compute the false alarm rate $\alpha$ according to $1 - \int_0^\infty \mathbb{P}(A(D') = o) \, do = 1 - \int_0^\infty \frac{1}{\sqrt{2 \log(1.25/\delta) \Delta Q/\epsilon}} \cdot 1 - \exp \left( - \frac{(o - Q(D'))^2}{4 \log(1.25/\delta) \Delta Q/\epsilon} \right) \, do$, which is $1 - \Phi \left( \frac{\theta - Q(D')}{\sqrt{2 \log(1.25/\delta) \Delta Q/\epsilon}} \right)$, i.e., $\theta = \Phi^{-1} \left( 1 - \alpha \right) \sqrt{2 \log(1.25/\delta) \Delta Q/\epsilon} + Q(D')$, where $\Phi(\cdot)$ is the cumulative dis-
tribution probability (CDF) of the standard normal distribution. Then, the threshold \( \lambda \) for the likelihood ratio can be computed as \( \exp \left( \frac{\Delta Q(2\theta - Q(D) - Q(D'))}{4 \log(1.25/\delta) \Delta Q / \epsilon} \right) \) and the true detection rate can be computed according to \( P_{TD} = 1 - \int_0^\infty \mathbb{P}(A(D) = o) d\alpha = 1 - \Phi \left( \frac{\theta - Q(D)}{\sqrt{2 \log(1.25/\delta) \Delta Q / \epsilon}} \right) \).

For a given false alarm rate \( \alpha \), we can uniquely determine the threshold of the likelihood ratio \( \lambda \), the threshold of the output \( \theta \) and the true detection rate \( P_{TD} \). Since \( \theta \) can be any possible value of the private query result, we know that the Neyman-Pearson criterion is equivalent to setting a threshold \( \theta \) for the Gaussian mechanism which is of a simpler formulation.

According to Theorem 21 and Eq. 5.10, we can theoretically derive precision and recall of the adversary’s hypothesis testing for the Gaussian mechanism as

\[
\text{precision} = \frac{1 - \Phi \left( \frac{\theta - Q(D)}{\sigma} \right)}{2 - \Phi \left( \frac{\theta - Q(D)}{\sigma} \right) - \Phi \left( \frac{\theta - Q(D')}{\sigma} \right)}
\]

\[
\text{recall} = 1 - \Phi \left( \frac{\theta - Q(D)}{\sigma} \right)
\]

(5.14)

We further show the corresponding PR-relation of the adversary’s hypothesis testing in Figure 5.4(a) and Figure 5.4(b) with varying \( \epsilon \) and \( \delta \), respectively. We observe that the adversary’s performance of hypothesis testing is enhanced with an increasing value of \( \epsilon \) and \( \delta \). Comparing Figure 5.3(a) and Figure 5.4, we know that different mechanisms vary in their power of defending against adversaries’ hypothesis testing since their output distributions are different. Note that our approach can be generally applied to any DP mechanism (Figure 5.1), although we focus on LPM and GPM.
Next, the highest $F_{\beta\text{score}}$ of the adversary’s hypothesis testing under different values of privacy parameters $\epsilon, \delta$ can be directly derived from Eq. 5.14 as

$$F_{\beta\text{score}}^* = \max_{\theta} \frac{(1 + \beta^2) \left( 1 - \Phi \left( \frac{\theta - Q(D)}{\sigma} \right) \right)}{2 + \beta^2 - \Phi \left( \frac{\theta - Q(D)}{\sigma} \right) - \Phi \left( \frac{\theta - Q(D')}{\sigma} \right)}$$  \hspace{1cm} (5.15)

5.4 Quantification of DP under Auxiliary Information from the Adversary’s Hypothesis Testing

We now demonstrate how to control the adversary’s success rate in identifying a particular record with several important variations of the adversary’s belief including the input data’s prior distribution, record correlation and temporal correlation.

5.4.1 Quantification of DP under Prior Distribution

Let us first consider an adversary with known prior distribution of the input data. Although such an adversary is not explicitly considered in conventional DP frameworks\textsuperscript{4}, we still analyze this adversary’s inference for sensitive information in a particular record as an interesting and practical case study. In some scenarios, the adversary’s prior is non-uniform, which will result in a different decision rule. Similar to our analysis in Section 5.3.2\textsuperscript{5}, we still consider a binary hypothesis testing problem where the adversary aims to distinguish the two neighboring databases in Eq. 5.12.\textsuperscript{4}

\textsuperscript{4}DP guarantees are not influenced by the prior distribution of the input data.
Next, we quantify the hypothesis testing of the adversary to distinguish the two posterior distributions of $P(d_i = d_{i1}|A(D) = o, d_{-i})$, $P(d_i = d_{i2}|A(D) = o, d_{-i})$. According to Bayes’ rule, we have

$$P(d_i = d_{i1}|A(D) = o, d_{-i}) = \frac{P(A(D) = o|d_{-i}, d_i = d_{i1})P(d_i = d_{i1})}{P(A(D) = o|d_{-i})}.$$  

Then, we get

$$P(d_i = d_{i1}|A(D) = o, d_{-i}) = \frac{P(A(D) = o)P(d_i = d_{i1})}{P(A(D) = o)P(d_i = d_{i1}) + P(A(D') = o)P(d_i = d_{i2})} \quad (5.16)$$

Based on Eq. 5.16, we know that the adversary’s hypothesis testing under prior distribution is equivalent to distinguishing the two probabilities of $P(A(D) = o)P(d_i = d_{i1})$ and $P(A(D') = o)P(d_i = d_{i2})$. Figure 5.5(a) shows the hypothesis testing procedure of the adversary, where $\theta$ is the decision threshold on the noisy query outputs and the adversary’s decision rule is $o \gt d_{i1} \lt \theta$. We further define the coefficient of prior distribution $\rho_p$ as $\rho_p = 1 - \min_{d_{i1}, d_{i2}} \frac{P(d_i = d_{i1})}{P(d_i = d_{i2})}$, where $\rho_p \in [0, 1]$. $\rho_p = 0$ corresponds to the scenario where the adversary has no knowledge about the prior distribution and thus makes the assumption of uniform distribution (the same as in Section 5.3.1). Combining Eq. 5.10 with Eq. 5.16 we can derive precision, recall as

$$\text{precision} = \frac{P(d_i = d_{i1})RSR}{P(d_i = d_{i1})RSR + P(d_i = d_{i2})GSR} = \frac{1}{1 + (1 - \rho_p)^{GSR/RSR}},$$

$$\text{recall} = RSR$$  

From Eq. 5.17, we know that precision is increased while recall is kept unchanged for adversaries knowing prior distribution of the input data. We further show this enhanced PR-relation in Figure 5.5(b) by setting $\rho_p = 0.2$ for instance, as a comparison to Figure 5.3(a) (corresponding to $\rho_p = 0$).
Figure 5.5. (a) Hypothesis testing and (b) PR-relation of detecting a particular record for adversaries with access to the prior distribution of the input data ($\rho_p = 0.2$).

Furthermore, we theoretically derive the highest $F_{\beta\text{score}}$ of the adversary’s hypothesis testing under different values of $\epsilon$ and $\rho_p$ as

$$F_{\beta\text{score}}^* = \begin{cases} 1 + \frac{\beta^2}{2 + \beta^2 - \rho_p}, & \epsilon < \log\left(1 + \frac{\beta^2}{1 - \rho_p}\right) \\ \frac{(1 + \beta^2)(\sqrt{1 + \frac{4\beta^2\epsilon}{1 - \rho_p}} - 1)}{(1 + \beta^2)\sqrt{1 + \frac{4\beta^2\epsilon}{1 - \rho_p}} - 1 + \beta^2}, & \epsilon \geq \log\left(1 + \frac{\beta^2}{1 - \rho_p}\right) \end{cases} \quad (5.18)$$

The detailed proof is deferred to the Appendix. Comparing Eq. 5.18 and Eq. 5.11, we know that 1) the adversary achieves an improved hypothesis testing by possessing auxiliary information of prior distribution and 2) a larger value of $\rho_p$ results in a higher confidence for the adversary to select the correct value of $d_i$ (with higher
Therefore, we conclude that the prior distribution of the input data should be considered when trying to select a proper $\epsilon$ in practice.

5.4.2 Quantification of DP under Record Correlation

Records in real world data often exhibit inherent dependencies or correlations. Handling correlated records is thus a significant problem, which has been demonstrated in previous work \cite{23, 71, 114, 121, 131, 133, 139}. Tschantz, Sen and Datta in \cite{121} investigate an interesting causal view of DP as limiting the effect of a single data point without independence assumptions, in order to resolve the confusion in prior work about DP under correlated data. Here, we will show the enhanced hypothesis testing of the adversary who has access to the correlation relationship across records. Note that we now consider the bounded case since the presence/absence of a record in the unbounded case has no relation with the correlation among records. This is the same for correlation across time which will be discussed in Sections 5.4.3.

Let us consider a general setting where the adversary aims to infer the value of $d_i$ while having access to values of all the other records $d_{-i}$ and the relationship between $d_i$ and its correlated records $d_{c1}, d_{c2}, \cdots$. Therefore, the adversary’s hypothesis testing tries to distinguish the two posterior probabilities of $P(d_i = d_{i1}|A(D) = o, d_{-i})$ and $P(d_i = d_{i2}|A(D) = o, d_{-i})$. Define $P_{i1}^c = P(d_i = d_{i1}|d_{c1}, d_{c2}, \cdots)$ and $P_{i2}^c = P(d_i = d_{i2}|d_{c1}, d_{c2}, \cdots)$. According to Bayes’ rule, we can derive

$$P(D_i = d_{i1}|A(D) = o, D_{c1} = d_{c1}, D_{c2} = d_{c2}, \cdots)$$

$$= \frac{P(A(D) = o|D_i = d_{i1}, D_{c1} = d_{c1}, D_{c2} = d_{c2}, \cdots)P_{i1}^c}{P(A(D) = o|D_{c1} = d_{c1}, D_{c2} = d_{c2}, \cdots)}$$

(5.19)
Then, we know

\[
P(d_i = d_{i1}|A(D) = o, d_{-i}) = \frac{P(A(D) = o)P_{c1}^e}{P(A(D') = o)P_{c2}^e}
\]  

(5.20)

Based on Eq. 5.20, we know that the adversary’s hypothesis testing under record correlation is equivalent to distinguishing the probabilities of \(P(A(D) = o)P_{c1}\) and \(P(A(D') = o)P_{c2}\) as shown in Figure 5.6(a). Let us further define the coefficient of record correlation \(\rho_c\) as

\[
\rho_c = 1 - \min_{d_{i1}, d_{i2}, \ldots} \frac{P(d_i = d_{i1})}{P(d_i = d_{i1}|d_{i1}, d_{i2}, \ldots)}, \quad \text{where } \rho_c \in [0, 1], \quad \text{and } \rho_c = 0 \text{ corresponds to the scenario of independent records.}
\]

Combining Eq. 5.10 and Eq. 5.20, we can compute the precision and recall for this adversary’s hypothesis testing as

\[
\text{precision} = \frac{P_{c1}^e \text{RSR}}{P_{c1}^e \text{RSR} + P_{c2}^e \text{GSR}} = \frac{1}{1 + (1 - \rho_p - \rho_c(2 - \rho_p)) \frac{\text{GSR}}{\text{RSR}}},
\]

(5.21)

\[
\text{recall} = \text{RSR}.
\]

Eq. 5.21 holds based on the fact that \(\max \frac{P_{c1}^e}{P_{c2}^e} = \max \frac{P_{c1}^e}{1 - \rho_{p} - \rho_{c}(2 - \rho_{p})} = \frac{1}{1 - \rho_{p} - \rho_{c}(2 - \rho_{p})}\). From Eq. 5.21, we know that precision is increased under the same level of recall for adversaries that possess record correlation of the input data. We show the enhanced PR-relation in Figure 5.6(b) by setting \(\rho_p = 0.2, \rho_c = 0.1\) for example. Furthermore, we theoretically derive the relationship between the highest \(F_{\beta\text{score}}\) and privacy budget \(\epsilon\), coefficient of prior distribution \(\rho_p\), coefficient of record correlation \(\rho_c\) as

143
The corresponding proof is deferred to the Appendix. From Eq. 5.22 we know that $F_{\beta_{score}}^*$ is further improved by record correlation accessible to the adversary, meaning that the adversary has more confidence to detect the true value of $d_i$. Therefore, the correlation across records should also be taken into consideration when selecting appropriate values of $\epsilon$ in practice.

5.4.3 Quantification of DP under Temporal Correlation

Temporal dynamics exist naturally in information networks, e.g., social network, mobility data, health records, etc. For instance, users’ location data which is provided to location-based services or applications are usually temporally correlated. Location-based social networks allow users to share locations with friends, to find friends, and to provide recommendations about points of interest based on their locations. Yet, individual privacy has been a major obstacle to data sharing. Many privacy frameworks including DP schemes do not explicitly incorporate such dynamics. Most of current perturbation mechanisms only consider static scenarios or perturb the location at single timestamps without considering temporal correlations of a moving user’s data. Even advanced variants of DP frameworks such as dependent differential privacy [83] only considers correlation among records in a single static database. Xiao et al. [132] and Cao et al. [16] consider temporal correlation across a single user’s data instead of under multiple users’ database. In practice, a time-series of users’ data may need
Let us consider a general temporal setting where the adversary aims to infer the value of $d_i^t$ at timestamp $t$ while having access to the values of all the other records $d_{-i}^{[t]}$, $d_i^{[t-1]}$ and the relationship between $d_i^t$ and its correlated records (across time and within this timestamp) $d_i^{[t-1]}$, $d_{c1}^{[t]}$, $d_{c2}^{[t]}$, $\ldots$. Similar to Sections 5.4.1, 5.4.2, the adversary aims to distinguish two posterior probabilities of $P(d_i^t = d_{i1}|A(D) = o, d_{-i}^{[t]}$, $d_i^{[t-1]})$ and $P(D_i^t = d_{i2}|A(D) = o, d_{-i}^{[t]}$, $d_i^{[t-1]})$. Defining $P_{i1}^t = P(d_i^t = d_{i1}|d_{i}^{[t-1]}, d_{c1}^{[t]}, d_{c2}^{[t]}, \ldots)$ and $P_{i2}^t = P(d_i^t = d_{i2}|d_{i}^{[t-1]}, d_{c1}^{[t]}, d_{c2}^{[t]}, \ldots)$.

\footnote{Here, $[t]$ represents timestamps from 1 to $t$.}

Figure 5.6. (a) Hypothesis testing and (b) PR-relation of detecting a particular record for adversaries with access to record correlation ($\rho_p = 0.2, \rho_c = 0.1$). To be published to enable real-world applications while satisfying rigorous privacy guarantees.
According to Bayes’ rule, we have

\[
P(D_i^t = d_{i1} | \mathcal{A}(D) = o, D_{i1}^{t-1} = d_i, D_{c1}^{t} = d_{c1}, D_{c2}^{t} = d_{c2}, \cdots) = \frac{P(\mathcal{A}(D) = o | D_i^t = d_{i1}, D_{i1}^{t-1} = d_i, D_{c1}^{t} = d_{c1}, D_{c2}^{t} = d_{c2}, \cdots) P_{i1}^t}{P(\mathcal{A}(D) = o | D_{i1}^{t-1} = d_i, D_{c1}^{t} = d_{c1}, D_{c2}^{t} = d_{c2}, \cdots)}.
\]  
(5.23)

Therefore, we can derive

\[
\frac{P(d_i^t = d_{i1} | \mathcal{A}(D) = o, d_i^{t-1}, d_{i2}^{t-1})}{P(d_i^t = d_{i2} | \mathcal{A}(D) = o, d_i^{t-1}, d_{i2}^{t-1})} = \frac{P(\mathcal{A}(D) = o) P_{i1}^t}{P(\mathcal{A}(D') = o) P_{i2}^t}.
\]  
(5.24)

From Eq. 5.24, we know that the adversary’s hypothesis testing under temporal dynamics is equivalent to distinguishing the two probabilities of \(P(\mathcal{A}(D) = o) P_{i1}^t\) and
P(A(D') = o)P_{t_2}' as shown in Figure 5.7(a). Let us define the coefficient of temporal correlation as \( \rho_t = 1 - \min_{d_1, d_1[t-1], d_1[t+1], \cdots} \frac{P(d_i = d_{i_1} | d_{i_2}, \cdots)}{P(d_i = d_i[t-1] | d_i[t+1], \cdots)} \), where \( \rho_t \in [0, 1] \) and \( \rho_t = 0 \) corresponds to the static scenario. Therefore, we can compute the precision and recall for this adversary’s hypothesis testing as

\[
\text{precision} = \frac{P_{t_1}' RSR}{P_{t_1}' RSR + P_{t_2}' GSR} = \frac{1}{1 + (1 - \rho_p - (2 - \rho_p)(\rho_c + \rho_t(1 - \rho_c))) GSR}, \quad (5.25)
\]

\[
\text{recall} = RSR.
\]

Eq. 5.25 holds since \( \max_{P_{t_1}'} \frac{P_{t_1}'}{P_{t_2}'} = \max_{P_{t_1}'} \frac{P_{t_1}'}{1-P_{t_1}'} = \frac{1}{1-\rho_p-(2-\rho_p)(\rho_c+\rho_t(1-\rho_c))}. \) From Eq. 5.25, we know that precision is increased under the same level of recall when the adversary has access to the temporal correlation of the input data, resulting in a better PR-relation compared to the static scenario. We show the enhanced PR-relation in Figure 5.7(b) by setting \( \rho_p = 0.2, \rho_c = 0.1, \rho_t = 0.1 \) for example. Furthermore, we theoretically compute the highest \( F_{\beta\text{score}} \) under given values of privacy budget \( \epsilon \), coefficient of prior distribution \( \rho_p \), coefficient of record correlation \( \rho_c \) and coefficient of temporal correlation \( \rho_t \) as

\[
F^*_{\beta\text{score}} = \begin{cases} 
\frac{1 + \beta^2}{2 + \beta^2 - \rho_p - (2 - \rho_p)(\rho_c + \rho_t(1 - \rho_c))}, & \epsilon < \epsilon(\rho_p, \rho_c, \rho_t) \\
(1 + \beta^2)\left(\sqrt{1 + \frac{4\beta^2 \epsilon}{1-\rho_p-(2-\rho_p)(\rho_c+\rho_t(1-\rho_c))}} - 1\right) - 1, & \epsilon \geq \epsilon(\rho_p, \rho_c, \rho_t) 
\end{cases} \quad (5.26)
\]

where \( \epsilon(\rho_p, \rho_c, \rho_t) = \log(1 + \frac{\beta^2}{1-\rho_p-(2-\rho_p)(\rho_c+\rho_t(1-\rho_c))}) \) and the corresponding proof is deferred to the Appendix. From Eq. 5.26, we know that the temporal correlation can benefit the adversary’s hypothesis testing to achieve an enhanced \( F^*_{\beta\text{score}} \). Therefore, the correlation of the input data across time should be considered in selecting appropriate privacy parameters of privacy preserving mechanisms.
Summary for the Quantification of DP under Auxiliary Information: Figure 5.8 shows the highest $F_{\beta \text{score}}$ (setting $\beta = 1$) varying with $\epsilon$ under different auxiliary information. We can observe that the adversary can infer more information of the input data with more auxiliary information (higher values of $\rho_p, \rho_c, \rho_t$). This property can also be explained by using conditioning always reduces entropy (uncertainty) in information theory [25]. Therefore, choosing a proper privacy budget $\epsilon$ needs more careful consideration when designing privacy-preserving mechanisms against adversaries who have access to these auxiliary information.

5.5 Quantification of Other Privacy Notions from the Adversary’s Hypothesis Testing

In this section, we systematically compare several existing statistical privacy frameworks from the perspective of the adversary’s hypothesis testing, including Pufferfish privacy [72], Blowfish privacy [58], dependent differential privacy [83], membership privacy [81], inferential privacy [49], and mutual-information based differential privacy [26] (detailed definitions are deferred to the Appendix). Our analysis can deepen
the understanding of these privacy notions as well as guide the design of their deployment in real world applications.

5.5.1 Qualitative Comparison of Different Privacy Metrics

From the perspective of adversary’s hypothesis testing, the adversary aims to distinguish between two neighboring databases from the noisy outputs satisfying different privacy metrics (recall Eqs. 5.1, 5.3, 5.12). We thus compare the definitions of neighboring databases in various privacy notions as shown in Figure 5.9. The neighboring databases in DP [34] considers the change of only one record in the database. Pufferfish privacy [72] aims to protect any potential secret of the database and Blowfish privacy [58] is a special class of privacy notions in the Pufferfish framework. The neighboring databases in Blowfish privacy with count query constraint and marginal constraint [58] can generally consider all the possible records' differences in the databases. Dependent differential privacy [83] and inferential privacy [49] aim to protect a particular record while taking its correlation with other records into consideration, therefore their neighboring databases are generated by the direct change of one record followed by possible changes of its correlated records. Furthermore, we analyze the relationship between these two notions. According to Bayes’ analysis in Eq. 5.4, we know inferential privacy that requires \[ \max_{D, D'} \frac{P(D = D'|A(D) = a)}{P(D = D'|A(D') = a)} \leq e^\epsilon \frac{P(D = D)}{P(D = D')} \] is equivalent to dependent differential privacy that requires \[ \max_{D, D'} \frac{P(A(D) = a)}{P(A(D') = a)} \leq e^\epsilon. \]

Note that membership privacy [81] and mutual-information differential privacy [26] are different from the above privacy metrics since their frameworks have not explicitly defined neighboring databases. We will show the relationship between them and all the other privacy metrics in the next subsection from the perspective of adversary’s hypothesis testing. Under the same privacy parameter, a privacy notion
that places less limitations to the neighboring databases can better restrict an adversary’s capability of performing hypothesis testing to infer sensitive information of an individual record. Therefore, using the same privacy budget, Blowfish privacy with count/marginal constraints can provide stronger defenses against the adversary’s hypothesis testing than dependent differential privacy and inferential privacy. Furthermore, these advanced variants of DP have explicitly taken the correlation among records into consideration, which are powerful in defending against adversaries who aim to utilize auxiliary knowledge to infer sensitive information (Section 5.4.2).

5.5.2 Quantitative Comparison of Different Privacy Metrics

For quantitative analysis, we obtain the main results for comparison of these privacy notions above in Theorem 22 below, and two propositions thereafter considering two special database scenarios (detailed proofs are deferred to the Appendix). Note that there is no general perturbation mechanism to achieve these advanced privacy notions. Therefore, it is difficult to numerically analyze the adversary’ hypothesis testing over these privacy notions as we did in the DP setting (recall Sections 5.3, 5.4).
Theorem 22. Privacy Comparison Main Result: Under the same performance of adversary’s hypothesis testing (denoted as $ht$ which can be PR-relation for instance), we have the following relationship for the privacy parameter $\epsilon$ used in different privacy notions.

\[
\epsilon_{BP}(ht) \geq \epsilon_{DDP}(ht) = \epsilon_{IP}(ht) \geq \epsilon_{DP}(ht)
\]
\[
\epsilon_{DDP}(ht) \leq 2\epsilon_{MP}(ht)
\]
\[
\epsilon_{MIDP}(ht) \leq \epsilon_{MP}(ht)
\]

where the subscripts $BP$, $DDP$, $IP$, $DP$, $MP$, $MIDP$ represent Blowfish privacy with count/marginal constraints, dependent differential privacy, inferential privacy, differential privacy, membership privacy, mutual-information differential privacy, respectively. Note that Blowfish privacy is a special subclass of the general Pufferfish framework that handles a set of deterministic constraints such as count/marginal constraints.

Theorem 22 states the relationship among values of $\epsilon$ in different privacy notions under the same level of PR-relation that can be achieved by the adversary. A privacy notion with a smaller $\epsilon$ in Theorem 22 is weaker in restricting an adversary’s capability of performing hypothesis testing to infer an individual record. Combining the qualitative analysis in Section 5.5.1 and the quantitative comparison shown in Theorem 22, we know that under the same level of PR-relation, a larger value of $\epsilon$ can be selected for a privacy notion with less restrictions in the definition of neighboring databases.

Based on Theorem 22, we further compare these privacy notions under two special data distributions: 1) independent records and 2) independent and uniform records, as shown in the following propositions.
Proposition 1. Privacy Comparison under Independent Records: If the individual records in the database are independent of each other, we have

\[ \epsilon_{BP}(ht) = \epsilon_{DDP}(ht) = \epsilon_{IP}(ht) = \epsilon_{DP}(ht) \]

\[ \epsilon_{DDP}(ht) \leq 2\epsilon_{MP}(ht) \]

\[ \epsilon_{MIDP}(ht) \leq \epsilon_{MP}(ht) \]

Proposition 2. Privacy Comparison under Independent and Uniform Records: If the individual records in the database are independent of each other, and each record is uniformly distributed, we have

\[ \epsilon_{BP}(ht) = \epsilon_{DDP}(ht) = \epsilon_{IP}(ht) = \epsilon_{DP}(ht) \]

\[ \epsilon_{DDP}(ht) \leq 2\epsilon_{MP}(ht) \]

\[ \epsilon_{MIDP}(ht) \leq \epsilon_{MP}(ht) \]

5.5.3 Chapter Summary

In this chapter, we investigate the state-of-the-art statistical privacy frameworks (focusing on DP) from the perspective of hypothesis testing of the adversary. We rigorously analyze the capability of an adversary for inferring a particular record of the input data using hypothesis testing. Our analysis provides a useful and interpretable guideline for how to select the privacy parameter \( \epsilon \) in DP, which is an important question for practitioners and researchers in the community. Our findings show that an adversary’s auxiliary information – in the form of prior distribution of the database, and correlation across records and time – indeed influences the proper choice of \( \epsilon \). Finally, our work systematically compares several state-of-the-art privacy notions from the perspective of adversary’s hypothesis testing and showcases their relationship with each other and with DP.
Chapter 6

Conclusion

In this thesis, we aim to overcome several challenges for current statistical data privacy mechanisms. In Chapter 3, we propose LinkMirage to effectively mediate privacy-preserving access to users’ social relationships. LinkMirage preserves key structural properties in the social topology while anonymizing intra-community and inter-community links. LinkMirage also provides rigorous guarantees for the anti-inference privacy, indistinguishability and anti-aggregation privacy, in order to defend against sophisticated threat models for both static and temporal graph topologies. LinkMirage significantly outperforms baseline static techniques in terms of both link privacy and utility, which have been verified both theoretically and experimentally using real-world datasets. LinkMirage enables the deployment of real-world social relationship based applications such as graph analytics, anonymity systems, and Sybil defenses while preserving the privacy of users’ social relationships.

Differential privacy provides a formal basis for expressing and quantifying privacy goals. To extend existing DP mechanisms to accommodate real world data dependence/correlation, we introduced a generalized dependent differential privacy frame-
work in Chapter 4 that incorporates probabilistic dependence relationship between data and provides rigorous privacy guarantees. We further propose a dependent perturbation mechanism and rigorously prove that it can achieve the privacy guarantees. Our evaluations over multiple large-scale real datasets and multiple query classes show that the dependent perturbation scheme performs significantly better than state-of-the-art approaches used for providing differential privacy.

Finally, we investigate the state-of-the-art statistical privacy frameworks (focusing on differential privacy) from the perspective of the adversary’s hypothesis testing in Chapter 5. We also consider adversaries who have access to three types of auxiliary information, namely prior distribution of the database, correlation across records and correlation across time. Furthermore, we systematically compare several state-of-the-art privacy notions from the perspective of hypothesis testing and analyze relationship among them and with differential privacy. Our analysis provides a useful and interpretable guideline for how to select the privacy parameter $\epsilon$ in differential privacy, which is an important question for practitioners and researchers.

6.1 Opportunities of Future Work

Although this dissertation has tackled several important problems related to designing effective statistical privacy frameworks and mechanisms, there remain many extensions and new directions for future research.

**Extending LinkMirage System to Broader Applications:** In our privacy analysis in Chapter 3, we consider the worst-case adversary who knows the entire social link information except one link, which conservatively demonstrates the superiority of our algorithm over the state-of-the-art approaches. LinkMirage benefits many appli-
cations that depend on graph-theoretic properties of the social graph (as opposed to the exact set of edges). This also includes recommendation systems and E-commerce applications. While our theoretical analysis of LinkMirage relies on undirected links, the obfuscation algorithm itself can be generally applied to directed social networks. Furthermore, our underlying techniques have broad applicability to domains beyond social networks, including communication networks and web graphs.

**Generalizing DDP to Accommodate Temporal Dependence and Other Practical Issues:** Our dependent differential privacy framework in Chapter 4 can also accommodate other dependent or correlated relationships such as temporal correlations across a time series of dataset, which opens up an interesting future research direction.

To form a deeper understanding of our dependent differential privacy, it will be interesting to explore the application of standard concepts in differential privacy to our framework, such as local sensitivity, smooth sensitivity [43].

One limitation of our work is that the dependence coefficient $\rho_{ij}$ is exactly known to both the adversary and the DPM designer. How to accurately compute the dependence coefficient and deal with the underestimation of $\rho_{ij}$ (as we discussed in Section 4.5.4.3) would be an interesting future work (note that the overestimation of $\rho_{ij}$ continues to provide rigorous DDP guarantees).

**Broadening Our Quantification Guidelines to Other Scenarios:** Our analysis in Chapter 5 focuses on the popular LPM-based DP mechanisms, based on which we illustrate how hypothesis testing can be used for the selection of privacy parameters. We have shown the generality of our approach by applying it to the Gaussian perturbation mechanism in Section 5.3.3. Investigating how to generalize our analysis to a broader range of privacy mechanisms and metrics such as the exponential mechanism,
randomized response, local DP and geo-indistinguishability \cite{6} could be interesting future directions.

In our work, we consider the adversary who aims to infer the presence/absence of any particular record (for unbounded DP) or the true value of a record (for bounded DP), which is the standard adversary considered in DP framework. In practice, the adversary may be more interested in some aggregate statistics of the record, for instance, whether the value of the record $d_i$ is higher than a given value $\gamma$. Under this scenario, the two hypotheses of the adversary can be constructed as $h_0 : d_i > \gamma, \ h_1 : d_i \leq \gamma$ and then similar analysis in Sections \cite{5.3, 5.4} can be conducted for implementing hypothesis testing. We will study the hypothesis testing of these adversaries in the future.

Our analysis considers adversaries with accurate auxiliary information of the prior distribution and correlation across records/time of the input database. In practice, it can be challenging for defenders to have an accurate estimate of the adversary’s auxiliary information. Therefore, investigating the capability of adversary’s hypothesis testing with approximate auxiliary information could be another interesting future work.

Motivated by composition properties of DP \cite{48, 67, 90}, it is interesting to investigate the composability of our analysis across different privacy mechanisms and explore tighter composition properties under specific mechanisms similar to \cite{67} in the future.

Ultimately, this dissertation advocates for a stronger integration of statistical privacy technologies in real-world applications. Taking into account privacy as an important design goal in addition to the conventional goals of enhancing utility performance of data analytics can allow the discipline of data science to reach its true potential.
Appendix A

Utility Analysis of LinkMirage System

A.1 Proof for Relating Utility Distance with Structural Metrics

From the definition of total variation distance, we have

$$\|P_{\mathbf{r}}(G'_t) - \pi\|_{TV} + \|P_{\mathbf{r}}(G_t) - \pi\|_{TV} \geq \|P_{\mathbf{r}}(G'_t) - P_{\mathbf{r}}(G)\|_{TV}.$$  

Taking the maximum over all vertices, we have

$$\max \|P_{\mathbf{r}}(G'_t) + \pi\|_{TV} + \max \|P_{\mathbf{r}}(G_t) - \pi\|_{TV} \geq \max \|P_{\mathbf{r}}(G'_t) - P_{\mathbf{r}}(G)\|_{TV} \quad (A.1)$$

Therefore, for $t \geq \tau_G(\epsilon)$, we have

$$\max \|P_{\mathbf{r}}(G'_t) - \pi\|_{TV} \geq \max \|P_{\mathbf{r}}(G'_t) - P_{\mathbf{r}}(G)\|_{TV} + \max \|P_{\mathbf{r}}(G_t) - \pi\|_{TV}$$

$$\geq \sum_{v=1}^{\|V_t\|} \frac{\|P_{\mathbf{r}}(G'_t) - P_{\mathbf{r}}(G)\|_{TV} - \pi\|_{TV}}{|V_t|} - \epsilon \quad (A.2)$$

$$= UD(G_t, G'_t, \tau_G(\epsilon)) - \epsilon$$
Furthermore, we obtain

$$\tau_{G_t'} (UD(G_t, G_{t}', \tau_{G_t} (\epsilon)) - \epsilon) \geq \tau_{G_t} (\epsilon) \quad (A.3)$$

It is known that the second largest eigenvalue modulus is related to the mixing time of the graph as $$\tau_{G_t} (\epsilon) \leq \frac{\log n + \log \frac{1}{\epsilon}}{1 - \mu_{G_t}}$$. From this relationship, we can bound the SLEM in terms of the mixing time as $$1 - \frac{\log n + \log \frac{1}{\tau_{G_t} (\epsilon)}}{\tau_{G_t} (\epsilon)} \leq \mu_{G_t}$$. Replacing $$\epsilon$$ with $$UD(G_t, G_{t}', \tau_{G_t} (\epsilon)) - \epsilon$$, we have

$$1 - \frac{\log n + \log \frac{1}{UD(G_t, G_{t}', \tau_{G_t} (\epsilon)) - \epsilon}}{\tau_{G_t} (UD(G_t, G_{t}', \tau_{G_t} (\epsilon)) - \epsilon)} \leq \mu_{G_t}' \quad (A.4)$$

Finally, we leverage $$\tau_{G_t}' (UD(G_t, G_{t}', \tau_{G_t} (\epsilon) - \epsilon)) \geq \tau_{G_t} (\epsilon)$$ in the above equation, to obtain $$\mu_{G_t}' \geq 1 - \frac{\log n + \log \frac{1}{UD(G_t, G_{t}', \tau_{G_t} (\epsilon)) - \epsilon}}{\tau_{G_t} (\epsilon)}$$.

### A.2 Proof of the Upper Bound for the Utility Distance

We first introduce some notations and concepts for describing our proof clearly. We consider two perturbation methods in the derivation process below. The first method is our dynamic perturbation method, which takes the graph evolution into consideration. The second method is the intermediate method, where we only implement dynamic clustering without the step of selective perturbation. That is to say, we cluster $$G_t$$, then perturb each community by the static method and each inter-community subgraphs by randomly connecting the marginal nodes, independently. We denote the perturbed graphs corresponding to the dynamic, the intermediate method by $$G_{t}', G_{t}'i$$ respectively. Similarly, we denote the perturbed TPM for the two approaches by $$P_{t}', P_{t}'i$$. For simplicity, we partition the proof into two stages. In the first stage, we
derive the UD upper bound for the intermediate perturbation method. In the second stage, we derive the relationship between $G_t^{i,j}$ and $G'_t$. Results from the two stages can be combined to find the upper bound for the utility distance of LinkMirage. Denoting the communities as $C_1, C_2, \cdots, C_{K_t}$ and the inter-community subgraphs as $C_{12}, C_{13}, \cdots$, we have

$$
\|P_t - P_t^{i,j}\|_{TV} = \begin{bmatrix}
P_{t(1,1)} - P'_{t(1,1)} & \cdots & P_{t(1,K_t)} - P'_{t(1,K_t)} \\
P_{t(2,1)} - P'_{t(2,1)} & \cdots & P_{t(2,K_t)} - P'_{t(2,K_t)} \\
\vdots & \ddots & \vdots \\
P_{t(K_t,1)} - P'_{t(K_t,1)} & \cdots & P_{t(K_t,K_t)} - P'_{t(K_t,K_t)}
\end{bmatrix}_{TV}
$$

$$= \frac{1}{|V_t|} \sum_{k=1}^{K_t} |V_t(k)| \|P_{t(k,k)} - P'_{t(k,k)}\|_{TV} + \frac{1}{|V_t|} \sum_{k,j=1, k \neq j}^{K_t} |E_t(k,j)| \|P_{t(k,j)} - P'_{t(k,j)}\|_{TV}$$

$$\leq \epsilon + \delta_t$$

Here, $\delta_t$ is the ratio cut of the graph $[3]$, and $\delta_t = |E_{t-in}|/|V_t| = \sum_{k,j=1, k \neq j}^{K_t} |E_t(k,j)|/|V_t|$. For arbitrary matrix $P$ and $Q$, we have $\|P - Q\|_{TV} \leq l \|P - Q\|_{TV}$. Combining the above results, we have $\text{UD}(G_t, G_t^{i,j}, l) \leq l \|P_t - P_t^{i,j}\|_{TV} \leq l (\epsilon + \delta_t)$. Then, we generalize the utility analysis of intermediate perturbation to our dynamic perturbation. Assume that there are $K_t^e$ out of $K_t$ clusters that are considered as changed, which would be perturbed independently, and $K_t^u$ out of $K_t$ clusters are considered as unchanged, i.e., their perturbation would follow the perturbation manner in $G'_t - 1$. To simplify derivation, we use $P_t(k)$ instead of $P_t(k,k)$ to represent the TPM of the
where $\epsilon_0$ denotes the threshold to classify a community as *changed* or *unchanged*. The last inequality comes from the fact that $\epsilon_0 \leq \epsilon$. Then, we can prove

$$
\text{UD}(G_t, G'_t, l) = \|P_t - P'_t\|_{TV} \\
\leq l\|P_t - P'_t\|_{TV} \leq l\|P_t - P'^{t,i}_t\|_{TV} + l(\epsilon + \delta_t)
$$

(A.7)

and

$$
\text{UD}(G_0, \cdots G_T, G'_0, \cdots G'_T, l) \leq \frac{1}{T+1} \sum_{t=0}^{T} 2l(\epsilon + \delta_t)
$$

(A.8)
Appendix B

Privacy Properties of Dependent Differential Privacy

B.1 Proof for Composition Theorems for DDP

Sequential Composition Theorem: For any sequence \( r \) of outcomes \( r_t \in \text{Region}(A_t) \) with the same dependence relationship \( R \), the probability of output \( r \) obtained from the sequence of \( A_t(D) \) is \( Pr(A(D) = r) = \prod_t Pr(A_t(D) = r_t) \). Applying the definition of DDP for each \( A_t \), we have

\[
\prod_t Pr(A_t(D) = r_t) \leq \prod_t Pr(A_t(D') = r_t) \times \prod_t \exp \left( \frac{\epsilon_t}{\Delta_{SQ}} \times |D - D'| \right) \leq Pr(A(D') = r) \times \exp \left( \sum_t \epsilon_t \right).
\]

Parallel Composition Theorem: For \( D \) and \( D' \), let \( D_t : D \cap D_t \) and \( D'_t = D \cap D_t \) with the same dependence relationship \( R \), for any sequence \( r \) of outcomes \( r_t \in \text{R}(A_t) \), the probability of output \( r \) obtained from the sequence of \( A_t(D) \) is \( Pr(A(D) = r) = \prod_t Pr(A_t(D_t) = r_t) \) Applying the definition of DDP for each \( A_t \), we have

\[
\prod_t Pr(A_t(D) = r_t) \leq \prod_t Pr(A_t(D'_t) = r_t) \times \prod_t \exp \left( \frac{\epsilon_t}{\Delta_{SQ}} \times |D_t - D'_t| \right) \leq Pr(A(D') = r) \times \exp \left( \max_t \epsilon_t \right) \leq Pr(A(D') = r) \times \exp \left( \max_t \epsilon_t \right).
\]
B.2 Proof for the Privacy Axioms of DDP

Transform Invariance Axiom:

\[
P(B(A(D)) = O|d_{i1}) = \sum_D P(B(A(D)) = O) P(D = D|d_{i1})
\]

\[
= \sum_D \sum_S P(B(S) = O) P(A(D = S|d_{i1})
\]

\[
\leq e^\epsilon \sum_D P(B(A(D)) = O) P(D = D|d_{i2})
\]

\[
= e^\epsilon P(B(A(D)) = O|d_{i2})
\]

(B.1)

Convexity Axiom:

\[
P(A^p(D) = S|d_{i1}) = p P(A_1(D) = S|d_{i1}) + (1 - p) P(A_2(D) = S|d_{i1})
\]

\[
\leq e^\epsilon p P(A_1(D) = S|d_{i1}) + e^\epsilon (1 - p) P(A_2(D) = S|d_{i1})
\]

(B.2)

\[
= e^\epsilon P(A^p(D) = S|d_{i2})
\]
Appendix C

Deriving $F^*_{\beta \text{score}}$ under Different Scenarios and Proving Theorem 22

C.1 Deriving $F^*_{\beta \text{score}}$ under Unbounded DP

According to the definition of $F_{\beta \text{score}}$, we know that maximizing $F_{\beta \text{score}}$ is equivalent to minimizing $\frac{GSR + \beta^2}{RSR}$ since $F_{\beta \text{score}} = \frac{1}{(1+\beta^2)^{\text{precision}}} + \frac{\beta^2}{(1+\beta^2)^{\text{recall}}} = \frac{1+\beta^2}{1+GSR+\beta^2}$. Then, we define $f = \frac{GSR + \beta^2}{RSR}$ and analyze $f$ by considering three different intervals of $\theta$ in $(-\infty, Q(D')], (Q(D'), Q(D)), [Q(D), +\infty)$, respectively.

1) For the first interval of $(-\infty, Q(D')]$, we have $f = \frac{GSR + \beta^2}{RSR} = \frac{1+\beta^2-0.5e^{-\frac{\theta-Q(D')}{\Delta Q}}}{1-0.5e^{-\frac{\theta-Q(D)}{\Delta Q}}}$. Then, we take the derivative of $f$ with respect to $\theta$ as

$$\frac{\partial f}{\partial \theta} = \frac{0.5e^{-\frac{\theta-Q(D')}{\Delta Q}}(1-0.5e^{-\frac{\theta-Q(D)}{\Delta Q}})}{(1-0.5e^{-\frac{\theta-Q(D)}{\Delta Q}})^2} + \frac{0.5e^{-\frac{\theta-Q(D)}{\Delta Q}}(1+\beta^2-0.5e^{-\frac{\theta-Q(D')}{\Delta Q}})}{(1-0.5e^{-\frac{\theta-Q(D)}{\Delta Q}})^2}$$

$$= \frac{0.5e^{-\frac{\theta-Q(D')}{\Delta Q}}((1+\beta^2)e^{-\epsilon} - 1)}{(1-0.5e^{-\frac{\theta-Q(D)}{\Delta Q}})^2}$$

(C.1)
Therefore, we have $\frac{\partial f}{\partial \theta} = 0$, i.e., the function $f$ increases monotonically with $\theta \in (-\infty, Q(D')]$, if $\epsilon < \log(1 + \beta^2)$. Otherwise, it decreases monotonically.

2) For the second interval of $(Q(D'), Q(D))$, we have $f = \frac{GSR+\beta^2}{RSR} = \frac{0.5e^{-\theta-Q(D')}}{1-0.5e^{-\theta-Q(D')}} + \beta^2$.

We then compute the derivative of $f$ with respect to $\theta$ as

$$\frac{\partial f}{\partial \theta} = -\frac{0.5e^{-\theta-Q(D')}}{\Delta Q}e^{-\theta-Q(D')}e^{-\theta-Q(D')} + \frac{0.5e^{-\theta-Q(D')}}{\Delta Q}(0.5e^{-\theta-Q(D')})^2 + \beta^2$$

$$= \frac{0.5e^{-\theta-Q(D')}}{\Delta Q}(e^{-\epsilon} - e^{-\theta-Q(D')})^2$$

When $\theta = Q(D)$, we have $\frac{\partial f}{\partial \theta} = \frac{0.5e^{-\theta-Q(D)}}{\Delta Q}(e^{-\epsilon} - e^{-\epsilon} + \beta^2) > 0$. When $\theta = Q(D')$, we have $\frac{\partial f}{\partial \theta} = \frac{0.5e^{-\theta-Q(D')}}{\Delta Q}(1 + \beta^2)e^{-\epsilon} - 1$. Therefore, we know that $\frac{\partial f}{\partial \theta} = 0$, i.e., the function $f$ increases monotonically with $\epsilon \in [Q(D'), Q(D)]$, if $\epsilon < \log(1 + \beta^2)$.

Otherwise, it decreases and then increases thus there is a minimum within this interval.

To solve for this minimum, we set the derivative to 0, i.e., $e^{-\epsilon} - e^{-\theta-Q(D')}/\Delta Qe^{-\epsilon} + \beta^2e^{-\theta-Q(D')}/\Delta Q = 0$ to obtain $\theta = \frac{\Delta Q}{\epsilon} \log \frac{1+\sqrt{1+4\beta^2e^{\epsilon}}}{2\beta^2} + Q(D')$. The corresponding minimum value for $F_{\beta\text{score}}$ can be computed as $\frac{(1+\beta^2)(1+4\beta^2e^{\epsilon})}{(1+\beta^2)(1+4\beta^2e^{\epsilon})-(1-\beta^2)e^{\epsilon}}$.

3) For the third interval $[Q(D), +\infty)$, we have $f = \frac{GSR+\beta^2}{RSR} = \frac{0.5e^{-\theta-Q(D')}/\Delta Q}{0.5e^{-\theta-Q(D')}/\Delta Q} + \beta^2$. We then take the derivative of $f$ with respect to $\theta$ as

$$\frac{\partial f}{\partial \theta} = -\frac{0.25e^{\theta-Q(D')}e^{-\theta-Q(D')}}{\Delta Q}(0.5e^{-\theta-Q(D')})^2 + \frac{0.5e^{-\theta-Q(D')}}{\Delta Q}(0.5e^{-\theta-Q(D')})^2 + \beta^2$$

$$= \frac{0.5\beta^2e^{-\theta-Q(D')}}{\Delta Q}(0.5e^{-\theta-Q(D')})^2$$

164
Therefore, we know that \( f \) increases monotonically with \( \theta \in [Q(D), +\infty) \). Combining the analysis for all the three intervals 1)-3), we obtain the highest \( F_{\beta \text{score}} \) as in Eq. 5.11.

### C.2 Deriving \( F_{\beta \text{score}}^* \) under Prior Distribution

Consider an adversary with access to the prior distribution of the input database. Since \( F_{\beta \text{score}} = \frac{1}{(1+\beta^2) \text{precision} + (1+\beta^2) \text{recall}} \), we know that maximizing \( F_{\beta \text{score}} \) is equivalent to minimizing \( \frac{(1-\rho_p) \text{GSR} + \beta^2}{\text{RSR}} \). Next, we define \( f = \frac{(1-\rho_p) \text{GSR} + \beta^2}{\text{RSR}} \) and analyze its property under three intervals.

1) For the first interval of \(( -\infty, Q(D') \]), we have

\[
f = \frac{(1-\rho_p) \text{GSR} + \beta^2}{\text{RSR}} = \frac{1-\rho_p + \beta^2 - 0.5(1-\rho_p)e^{-(1-0.5\rho_p)e^{\frac{\theta-Q(D')}{\Delta Q}}}}{1-0.5e^{\frac{\theta-Q(D)}{\Delta Q}}}
\]  

(C.4)

Next, we take the derivative of \( f \) as

\[
\frac{\partial f}{\partial \theta} = \frac{-0.5(1-\rho_p)e^{\frac{\theta-Q(D')}{\Delta Q}}(1-0.5e^{-\frac{\theta-Q(D)}{\Delta Q}}) + 0.5e^{\frac{\theta-Q(D)}{\Delta Q}}(1-\rho_p + \beta^2 - 0.5(1-\rho_p)e^{\frac{\theta-Q(D')}{\Delta Q}})}{(1-0.5e^{\frac{\theta-Q(D)}{\Delta Q}})^2}
\]

\[
= \frac{0.5e^{\frac{\theta-Q(D')}{\Delta Q}}((1-\rho_p + \beta^2)e^{-\epsilon} - (1-\rho_p))}{(1-0.5e^{\frac{\theta-Q(D)}{\Delta Q}})^2}
\]

(C.5)

Therefore, we have \( \frac{\partial f}{\partial \theta} \mid_{\theta=Q(D')} > 0 \), i.e., the function \( f \) increases monotonically for \( \theta \in (-\infty, Q(D')] \), if \( \epsilon < \log(1 + \frac{\beta^2}{1-\rho_p}) \). Otherwise, it decreases monotonically.
2) For the second interval of \((Q(D'), Q(D))\), we have

\[
f = \frac{(1 - \rho_p)GSR + \beta^2}{RSR} = \frac{0.5(1 - \rho_p)e^{-\frac{\theta_Q(D')}{\Delta Q}} + \beta^2}{1 - 0.5e^{-\frac{\theta_Q(D')}{\Delta Q}}}
\]  

(C.6)

Next, we take the derivative of \(f\) as

\[
\frac{\partial f}{\partial \theta} = \frac{-0.5(1-\rho_p)e^{-\frac{\theta_Q(D')}{\Delta Q}}(1 - 0.5e^{-\frac{\theta_Q(D')}{\Delta Q}})}{(1 - 0.5e^{-\frac{\theta_Q(D')}{\Delta Q}})^2} + \frac{0.5\epsilon e^{-\frac{\theta_Q(D')}{\Delta Q}}(0.5(1 - \rho_p)e^{-\frac{\theta_Q(D')}{\Delta Q}} + \beta^2)}{(1 - 0.5e^{-\frac{\theta_Q(D')}{\Delta Q}})^2}
\]

(C.7)

\[
\frac{\partial f}{\partial \theta} = \frac{0.5(1-\rho_p)e^{-\epsilon} - e^{-\frac{\theta_Q(D')}{\Delta Q}} + \beta^2}{(1 - 0.5e^{-\frac{\theta_Q(D')}{\Delta Q}})^2}
\]

For \(\theta = Q(D)\), we have \(\frac{\partial f}{\partial \theta = Q(D)} = \frac{0.5(1-\rho_p)e^{-\epsilon} - e^{-\epsilon} + \beta^2}{1-\rho_p}\) > 0. For \(\theta = Q(D')\), we have \(\frac{\partial f}{\partial \theta = Q(D')} = \frac{0.5(1-\rho_p)e^{-\epsilon} - (1 + \frac{\beta^2}{1-\rho_p})e^{-\epsilon} - 1}\). Therefore, we have \(\frac{\partial f}{\partial \theta = Q(D')} > 0\), i.e., the function \(f\) increases monotonically for \(\epsilon \in (Q(D'), Q(D))\), if \(\epsilon < \log(1 + \frac{\beta^2}{1-\rho_p})\).

Otherwise, it decreases and then increases thus there is a minimum within this interval.

To solve for the minimum, we set the derivative to 0, i.e., \(e^{-\epsilon} - e^{-\frac{\theta_Q(D')}{\Delta Q}} + \frac{\beta^2}{1-\rho_p} \cdot \frac{\theta_Q(D')}{\Delta Q} = 0\) to obtain \(\theta = \frac{\Delta Q}{\epsilon} \log\left(\frac{1-\rho_p}{1-\rho_p} \sqrt{1 + \frac{4\beta^2e^\epsilon}{1-\rho_p}} - 1\right) + Q(D')\). The corresponding

\[
F_{\beta score} = \frac{(1 + \beta^2)(\sqrt{1 + \frac{4\beta^2e^\epsilon}{1-\rho_p}} - 1)}{(1 + \beta^2)(\sqrt{1 + \frac{4\beta^2e^\epsilon}{1-\rho_p}} + 1 + \beta^2)}
\]

3) For the third interval of \([Q(D'), +\infty)\), we have

\[
f = \frac{(1 - \rho_p)GSR + \beta^2}{RSR} = \frac{0.5(1 - \rho_p)e^{-\frac{\theta_Q(D')}{\Delta Q}} + \beta^2}{0.5e^{-\frac{\theta_Q(D')}{\Delta Q}}} + \beta^2
\]

(C.8)
Then, we take the derivative of \( f \) as

\[
\frac{\partial f}{\partial \theta} = -0.25(1 - \rho_p)^2 e^{-\frac{\theta - Q(D')}{\Delta Q}} e^{-\frac{\theta - Q(D)}{\Delta Q} \epsilon} + \frac{0.5 \epsilon e^{-\frac{\theta - Q(D)}{\Delta Q} \epsilon}(0.5(1 - \rho_p) e^{-\frac{\theta - Q(D')}{\Delta Q} \epsilon} + \beta^2)}{(0.5 e^{-\frac{\theta - Q(D)}{\Delta Q} \epsilon})^2}
\]

\[
= \frac{0.5 \beta^2 e^{-\frac{\theta - Q(D)}{\Delta Q} \epsilon}}{(0.5 e^{-\frac{\theta - Q(D)}{\Delta Q} \epsilon})^2}
\]

Therefore, \( f \) increases monotonically for \( \theta \in (-\infty, Q(D')] \). Combining the analysis for all the three intervals, we can achieve \( F_{\beta\text{score}}^* \) as in Eq. 5.11.

C.3 Deriving \( F_{\beta\text{score}}^* \) under Record Correlation

The computation of the highest \( F_{\beta\text{score}} \) for adversaries under record correlation is similar to that of adversaries with prior distribution. By comparing Eq. 5.17 and Eq. 5.21 we can also simply replace \( \rho_p \) in Eq. 5.18 with \( \rho_p + \rho_c(2 - \rho_p) \) to obtain Eq. 5.22.

C.4 Deriving \( F_{\beta\text{score}}^* \) under Temporal Correlation

The computation of the highest \( F_{\beta\text{score}} \) for adversaries under temporal correlation is similar to that of adversaries with prior distribution and record correlation. By comparing Eq. 5.17 with Eq. 5.25 we can also simply replace \( \rho_p \) in Eq. 5.18 with \( \rho_p + (2 - \rho_p)(\rho_c + \rho_t(1 - \rho_c)) \) to obtain Eq. 5.26.
C.5 Proof for Theorem 22

Proof. The Blowfish framework [58], which is a subclass of the Pufferfish framework, allows user to specify adversarial knowledge about the database in the form of deterministic policy constraints. In the presence of general deterministic constraints, pairs of neighboring databases can differ in any number of tuples. The neighboring databases of dependent differential privacy differ in \( L \) tuples caused by one direct modification of one tuple. Therefore, under the same performance achieved by the adversary’s hypothesis testing, we know that \( \epsilon_{DDP}(ht) \leq \epsilon_{BP}(ht) \).

Next, from the definition of inferential privacy, we have

\[
\max_{D,D'} \frac{P(D \sim D) \cdot \text{A}(D) = o}{P(D \sim D')} \leq \max_{D,D'} \frac{P(D \sim D) \cdot \text{A}(D) = o}{P(D \sim D')} \leq e^\epsilon \quad \text{which is equivalent to dependent differential privacy that requires } \max_{D,D'} \frac{P(D \sim D) \cdot \text{A}(D) = o}{P(D \sim D')} \leq e^\epsilon. \]

Therefore, we have \( \epsilon_{DDP}(ht) = \epsilon_{IP}(ht) \) under the same hypothesis testing performance achieved by the adversary.

Furthermore, since the neighboring databases in DP only differ in one tuple, we know that \( \epsilon_{DP}(ht) \leq \epsilon_{DDP}(ht) \) under the same hypothesis testing achieved by the adversary.

Next, we consider membership privacy whose mathematical form is different from other privacy metrics. Membership privacy does not consider two neighboring databases, but only bounds the ratio between the posterior probability and the prior probability for any data record. Based on the definition of membership privacy and inferential privacy, we know that the membership privacy satisfying \( \exp(-\epsilon) \leq \frac{P(D \sim D) \cdot \text{A}(D) = o}{P(D \sim D)} \leq \exp(\epsilon) \) would lead to

\[
\max_{d_{i1}, d_{i2}} \frac{P(D \sim d_{i1}) \cdot \text{A}(D) = o}{P(D \sim d_{i2}) \cdot \text{A}(D) = o} \leq \exp(2\epsilon) \frac{P(D \sim d_{i1})}{P(D \sim d_{i2})} \quad \text{in inferential privacy (and thus dependent differential privacy). That is to say, } \epsilon\text{-membership privacy would lead to }
\]
2\(\epsilon\)-dependent differential privacy. Therefore, we have \(\epsilon_{\text{DDP}}(ht) \leq 2\epsilon_{\text{MP}}(ht)\) under the same performance of the adversary’s hypothesis testing.

Furthermore, membership privacy considers the worst-case difference between posterior probability and prior probability, and the mutual information based differential privacy considers the average difference between posterior probability and prior probability. Therefore, we have \(\epsilon\)-membership privacy would lead to \(\epsilon\)-mutual information based differential privacy. Under the same performance of the adversary’s hypothesis testing, we have \(\epsilon_{\text{MIDP}}(ht) \leq \epsilon_{\text{MP}}(ht)\). 

Bibliography


