

TOWARDS PRACTICING PRIVACY IN
SOCIAL NETWORKS

by

XIAO QIAN

(B.Sc., Beijing Normal University, 2009)

A THESIS SUBMITTED

FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

NUS GRADUATE SCHOOL FOR INTEGRATIVE
SCIENCES AND ENGINEERING

at the

NATIONAL UNIVERSITY OF SINGAPORE

2014

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Xiao Qian

Xiao Qian

August 13, 2014

Acknowledgments

“Two are better than one; because they have a good reward for their labour.”

— Ecclesiastes 4:9

I always feel deeply blessed to have Prof TAN Kian-Lee as my Ph.D. advisor. He is my mentor, not only in my academic journey, but also in spiritual and personal life. I am forever indebted to him. His gentle wisdom is always my source of strength and inspiration. He keeps exploring the research problems together with me, cherishes each work as his own. During my difficult time in research, he never let me feel alone and kept encouraging and supporting me. I am truly grateful for the freedom he gives in research, greatly touched by his sincerity, and deeply impressed by his consistency and humility in life.

I always feel extremely fortunate to have Dr. CHEN Rui as my collaborator. Working with him always brings me cheerful spirits. When I encounter difficulties in research, CHEN Rui’s insights always bring me sparkles, and help me in time to overcome the hurdles. I have also truly benefited from his sophistication in thoughts and succinctness in writing.

I would like to thank Htoo Htet AUNG for spending time to discuss with me and teach me detailed research skills, CAO Jianneng for teaching me the importance of perseverance in Ph.D., WANG Zhengkui for always helping me and giving me valuable suggestions, Gabriel GHINITA and Barbara CARMINATI for their kindness and gentle guidance in research. These people are the building blocks for my works in the past five years’ study.

I am very grateful to have A/Prof Roger ZIMMERMANN and A/Prof Stephane BRESSAN as my Thesis Advisory Committee members. Thanks for their precious time and constant help all these years. Moreover, I would also like to thank A/Prof Stephane BRESSAN for giving me opportunities to collaborate with his research group, especially with his student SONG Yi.

I am very thankful for my friends. They bring colors into my life. In particular, I would like to thank SHEN Yiyang and LI Xue for keeping me company during the entire duration of my candidature; GAO Song for his generous help and precious encouragement in times of difficulty; WANG BingYu and YANG Shengyuan for always being my joy. I would also like to thank my sweet NUS dormitory roommates,

together with all my lovely labmates in SOC database labs and Varese's research labs, especially CAO Luwen, WANG Fangda, ZENG Yong and KANG Wei. They are my trusty buddies and helping hands all the time. Special thanks to GAO Song, LIU Geng, SHEN Yiying and YI Hui for helping me refine this thesis.

I would also like to thank Lorenzo BOSSI for being there and supporting me, in particular for helping me with the software construction.

I would never finish my thesis without the constant support from my beloved parents, XIAO Xuancheng and JIANG Jiuhong. I always feel deeply fulfilled to see they are so cheerful even for very small accomplishments that I've achieved. Their unfailing love is a never-ending source of strength throughout my life.

Lastly, thank God for His words of wisdom, for His discipline, perfect timing and His sovereignty over my life.

Contents

Acknowledgments	i
Summary	vii
List of Tables	ix
List of Figures	xi
1 Introduction	1
1.1 Thesis Overview and Contributions	2
1.1.1 Privacy-aware OSN data publishing	2
1.1.2 Collaborative access control	6
1.1.3 Thesis Organization	7
2 Background and Related Works of OSN Data Publishing	9
2.1 On Defining Information Privacy	9
2.2 On Practicing Privacy in Social Networks	12
2.2.1 Applying k -anonymity on social networks	12
2.2.2 Applying anonymity by randomization on social networks . .	14
2.2.3 Applying differential privacy on social networks	16
3 LORA: Link Obfuscation by RAndomization in Social Networks	19
3.1 Introduction	19
3.2 Preliminaries	23
3.2.1 Graph Notation	23
3.2.2 Hierarchical Random Graph and its Dendrogram Representa- tion	23

3.2.3	Entropy	26
3.3	LORA: The Big Picture	27
3.4	Link Obfuscation by Randomization with HRG	29
3.4.1	Link Equivalence Class	29
3.4.2	Link Replacement	30
3.4.3	Hide Weak Ties & Retain Strong Ties	30
3.5	Privacy Analysis	31
3.5.1	The Joint Link Entropy	32
3.5.2	Link Obfuscation VS Node Obfuscation	35
3.5.3	Randomization by Link Obfuscation VS Edge Addition/Deletion	36
3.6	Experimental Studies	37
3.6.1	Datasets	37
3.6.2	Experimental Setup	37
3.6.3	Data Utility Analysis	38
3.6.4	Privacy Analysis	43
3.7	Summary	43
4	Differentially Private Network Data Release via Structural Inference	45
4.1	Introduction	45
4.2	Preliminaries	48
4.2.1	Hierarchical Random Graph	48
4.2.2	Differential Privacy	50
4.3	Structural Inference under Differential Privacy	51
4.3.1	Overview	51
4.3.2	Algorithms	52
4.4	Privacy Analysis	56
4.4.1	Privacy via Markov Chain Monte Carlo	56
4.4.2	Sensitivity Analysis	57
4.4.3	Privacy via Structural Inference	60
4.5	Experimental Evaluation	60
4.5.1	Experimental Settings	61
4.5.2	Log-likelihood and MCMC Equilibrium	61
4.5.3	Utility Analysis	63

4.6	Summary	67
5	Background and Related Works of OSN Collaborative Access Control	71
5.1	Enforcing Access Control in the Social Era	71
5.1.1	Towards large personal-level access control	72
5.1.2	Towards distance-based and context-aware access control	72
5.1.3	Towards relationship-composable access control	72
5.1.4	Towards more collective access control	73
5.1.5	Towards more negotiable access control	73
5.2	State-of-the-art OSN Access Control Strategies	74
6	Peer-aware Collaborative Access Control	77
6.1	Introduction	77
6.2	Representation of OSNs	80
6.3	The Big Picture	81
6.4	Player Setup	85
6.4.1	Setting I-Score	85
6.4.2	Setting PE-Score	87
6.5	The Mediation Process	88
6.5.1	An Example	88
6.5.2	The Mediation Engine	89
6.5.3	Constraining the I-Score Setting	92
6.6	Discussion	95
6.6.1	Configuring the set-up	95
6.6.2	Second Round of Mediation	97
6.6.3	Circle-based Social Network	99
6.7	User Interface	100
6.8	Summary	103
7	Conclusion and Future Directions	105
7.1	Towards Faithful & Practical Privacy-Preserving OSN data publishing	105
7.2	Integrating data-access policies with differential privacy	107
7.3	New privacy issues on emerging applications	108

Towards Practicing Privacy in Social Networks

by

Xiao Qian

Submitted to the
NUS Graduate School for Integrative Sciences and Engineering
on August 13, 2014,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy

Summary

Information privacy is vital for establishing public trust on the Internet. However, as online social networks (OSNs) step into literally every aspect of our life, they also further erode our personal privacy to an unprecedented extent. Today, network data releasing and inadvertent OSN privacy settings have become two main channels causing such privacy leakage. As such, there is an urgent need to develop practical privacy preservation techniques. To this end, this thesis studies the challenges raised in the above two settings and develops practical techniques for privacy-preservation for today's OSNs.

For the first setting, we investigate two widely-adopted privacy concepts for data publication, namely, anonymization and differential privacy. We utilize the *hierarchical random graph*(HRG) model to develop privacy preserving techniques to ground privacy from two disparate perspectives, one from anonymization and another from statistical disclosure control.

Specifically, we first show how HRG manifests itself as a promising structure that offers space for adding randomness to the original data while preserving good network properties. We illustrate how the best-fitting HRG structure can achieve anonymity via obfuscating the existence of links in the networks. Moreover, we formalize the randomness regarding such obfuscation using entropy, a concept from information theory, which quantifies exactly the notion of uncertainty. We also conduct experimental studies on real world datasets to show the effectiveness of this approach.

Next, rather than introducing randomness in the best-fitting HRG structure, we design a differentially private scheme that reaps randomness by sampling in the entire HRG model space. Compare to other competing methods, our sampling-based strategy can greatly reduce the added noise required by differential privacy. We formally prove that the sensitivity of our scheme is of a logarithmic order in the network's size. Empirical experiments also indicate our strategy can preserve network utility well while strictly controlling information disclosure in a statistical sense.

For the second setting, we attempt to solve an equally pressing emerging problem. In today's OSN sites, many content such as group photos and shared documents are co-owned by multiple OSN users. This prompts the need of a fast and flexible decision-making strategy for collaborative access control over these co-owned contents online. We observe that, unlike traditional cases where co-owners' benefits usually conflict with those of each other, OSN users are often friends and care for each other's

emotional needs. This in turn motivates the need to integrate such peer effects into existing collaborative access control strategies. In our solution, we apply game theory to develop an automatic online algorithm simulating an emotional mediation among multiple co-owners. We present several examples to illustrate how the proposed solution functions as a knob to coordinate the collective decision via peer effects. We also develop a Facebook app to materialize our proposed solution.

Thesis Supervisor: Tan Kian-Lee
Title: Professor

List of Tables

3.1	Network dataset statistics	38
6.1	Initial I-Scores with Method OO	88
6.2	Peer Effects Scores	89
6.3	I-Scores at Equilibrium with Method OO	89
6.4	Initial I-Scores with Method OC	93
6.5	I-Scores at Equilibrium with Method OC	93
6.6	PE-Scores before adjustment	96
6.7	PE-Scores after adjustment	96
6.8	Initial I-Scores in the extreme case	96
6.9	I-Scores at Equilibrium in the extreme case	97
6.10	Intercentrality Scores	98
6.11	Adjusted Initial I-Scores with Method OC	98
6.12	I-Scores at Equilibrium with Method OC in the Second Mediation	99

List of Figures

2-1	Timeline of Selected Works on Privacy-preserving Data Publishing . . .	13
3-1	An example of HRG model in [CMN08; CMN07].	25
3-2	Perturbed Graph & Node Generalization	30
3-3	Link Obfuscation VS Random Sparsification	36
3-4	Degree distribution	40
3-5	Shortest Path Distribution	40
3-6	Overlap percentage of top-k influential vertices	41
3-7	Mean absolute error of top-k vertices	41
3-8	Egocentric entropy	42
4-1	An example of the HRG model in [CMN08]	49
4-2	Three configurations of r 's subtrees [CMN08]	53
4-3	Gibbs-Shannon entropy and plot of Δu	59
4-4	Trace of log-likelihood as a function of the number of MCMC steps, normalized by n	62
4-5	Degree distribution	64
4-6	Shortest path length distribution	64
4-7	Overlaps of top- k vertices	65
4-8	Mean absolute error of top- k vertices	65
4-9	<i>polblogs</i> with <i>hrg</i> -0.3	66
4-10	<i>polblogs</i> with <i>hrg</i> -0.5	67
4-11	<i>wiki-Vote</i> with <i>hrg</i> -0.3	68
4-12	<i>wiki-Vote</i> with <i>hrg</i> -0.5	68
4-13	<i>ca-HepPh</i> with <i>hrg</i> -0.3	69
4-14	<i>ca-HepPh</i> with <i>hrg</i> -0.5	69

4-15	<i>ca-AstroPh</i> with <i>brg</i> -0.3	70
4-16	<i>ca-AstroPh</i> with <i>brg</i> -0.5	70
6-1	The CAPE Framework	83
6-2	Two Designs of Intensity Bar	87
6-3	Peer effects in OSN	89
6-4	CAPE-Login	101
6-5	CAPE-PEScores	101
6-6	CAPE-IScores	102
6-7	CAPE-Mediation Outcome	102

Chapter 1

Introduction

Information privacy, as it turns out, has now become the cornerstone of public trust on the Internet. Over the past decade, we have witnessed striking revelations of government surveillance over the Internet, countless lawsuits against big technology companies due to accidental leakage of user data, as well as unexpected embarrassment and harms caused by careless privacy setting in Facebook(e.g., wide circulation of personal photos than initially intended, online harassment and stalking powered by today's advanced searching engines like Facebook Graph Search). Perhaps without these incidents raised over the Internet, especially those in online social networks, we may never realize that privacy is so important and yet so fragile. As one of the fundamental human rights, privacy is now of utmost importance to us.

What makes privacy so difficult to protect today? One reason is that we are now more connected than ever. Statistics showed that online social networks(OSN) shrink our degree of separation in the world - from six degrees in the past to 4.74 degrees in the Internet today [Bac11]. As we connect to more people, we also open more channels that can leak our personal data, especially when we do not carefully pick our audience for what we share online. Secondly, as OSN media greatly enriches our ways of self-expression, they also advocate further disclosure of ourselves, from our words(text) to photos(images), from where we are(locations), whom we connect with (relationships), to what we like(wish list) and what we have bought(transaction records). This information contains great potential business opportunities and valuable research resources. Hence, many e-commerce companies, application developers and academic researchers crawl OSNs to collect huge amount of user data. However,

the personal information, once available to malicious attackers, is more than enough to uniquely identify a person. Thirdly, as all the information is stored online, users virtually do not have full control over their data. The data can be easily exposed and reproduced through, for instances, secret surveillance by government or data exchanges between companies. Lastly, even for the part that user can control, one cannot expect everyone to be an access control expert, bustling with endless maintenance tasks for the complicated OSN privacy settings.

Clearly, unrestrained collection of OSN data and careless privacy settings can put our privacy in serious jeopardy in the era of social media. Acknowledging that it is impossible for us to perfectly prevent privacy leakage today, we can, however, still push the boundaries for limiting such leakage, that is, put such leakage under control, limit unintended data access, and make precise identification difficult to achieve. These critical privacy issues, once solved, can have a profound impact on reforming data protection legislation and restoring the trust on the Internet. This thesis is dedicated to investigating a few new techniques to tackle such problems, aiming to offer new perspectives as well as technical tools for protecting an individual's privacy in OSNs.

1.1 Thesis Overview and Contributions

The thesis addresses problems raised as practicing privacy in social networks from two aspects. We first consider the problem of privacy-aware OSN data publishing. We will present one perturbation-based anonymization approach as well as one differentially private randomization strategy. Next, we will address another concern of OSN privacy protection from a complementary aspect, that is, facilitating individual users in configuring their privacy setting in OSN sites. In this part, we will mainly focus on the practical issues of applying access control techniques in a collaborative scenario.

1.1.1 Privacy-aware OSN data publishing

As OSN sites become prevailing worldwide, they also become invaluable data sources for many applications: personalized recommendation/services; targeted advertisements; knowledge discovery of human interaction at an unprecedented scale; vital channels connecting people in emergency and disasters like earthquake, terrorist attacks, etc.

In academics, in industry, and in numerous apps in app ecosystems(e.g. google play), we observe the increasing demands for much more broader OSN data sharing and data exchanges.

Despite many applications utilizing OSN data for good intentions, unrestrained collection of OSN data can seriously threaten individual’s privacy. For example, a great deal of details about government surveillance over the Internet had been revealed recently(e.g., PRISM¹). Even though this action is originally meant for national security, it, meanwhile, seriously undermines public trust. To restore user’s trust in OSNs, the leading companies, e.g., Facebook and Twitter, appeal together to the government for reforming privacy laws and regulating such surveillance². However, so far the legal definition of privacy still remains vague in concept. There is an urgent need to make the notion of privacy measurable, quantifiable and actionable, which is essential to make privacy protection operational in the juridical practice.

In this thesis, we will present two specific techniques for privacy-aware OSN data publishing. Most earlier notable works in this line employed *k-anonymity*, a privacy definition that requires the information for each person contained in the data to be indistinguishable from at least $k - 1$ individuals. This is based on the initial attempt to define privacy by considering it equivalent to preventing individuals from being re-identified. However, each of these works based on *k-anonymity* is only defined to satisfy an ad-hoc privacy measure. This means one method is only resilient to one specific type of attack, and hence would always be susceptible to new types of attacks.

Anonymity-based Data Publication

Our first contribution in this thesis is to adopt a random perturbation approach (another main branch of anonymity-based privacy methods) to achieve anonymity. In our works, we put our focus on protecting the existence of links in networks. We will show that, from information theory’s point of view, the proposed method can ground privacy via obfuscation, which can be accurately quantified by entropy. Briefly, we introduce a method that utilizes the *hierarchical random graph*(HRG) model to contextualize such obfuscation regarding link existence into the original network data. We will show how HRG manifests itself to be a promising structure that offers space

¹<http://www.cnn.com/2013/12/10/opinion/oppenheim-privacy-reform/index.html>

²<https://www.reformgovernmentsurveillance.com/>

for adding randomness in the original data while preserving good network properties. Briefly, we will illustrate how a best-fitting HRG can be used to recognize the set of substitute links, which can replace real links in the original network without greatly sacrificing the network’s global structure. Hence, instead of scrubbing the original network to rule out the data “finger-prints”(e.g. degree, neighborhood structure) from re-identification, the typical paradigm under *k-anonymity* framework, we can tailor the network regarding its own structure as carrying out perturbation to achieve link existence obscurity.

Furthermore, we formalize the notion of “link entropy” to quantify the privacy level regarding the existence of links in the network. We specifically present in details how to measure “link entropy” given a best-fitting HRG structure with regard to the original network. We also conduct experiments on four real-life datasets. Empirical results also show that, the proposed method allows a great portion of links to be replaced, which indicates the eligible perturbed network to release shall contain a significant amount of uncertainty concerning the existence of links. Results also show that the proposed method can still harvest good data-utility(e.g., degree distribution, shortest path length and influential nodes) after large numbers of edges being perturbed.

Differentially Private Data Publication

Despite many works on anonymity, subsequently, researchers began to realize that it can never provide full privacy guarantee in case of linkage attack. The reason is that, one can always anticipate, with sufficient auxiliary information, an attacker can always uniquely re-identify a person in OSN with the released dataset satisfying any privacy definition based on anonymity. To protect against linkage attack, *differential privacy*(DP) was introduced and has been widely adopted by researchers recently. Unlike anonymization methods, DP judges the data-releasing mechanism under consideration itself. More precisely, it measures the privacy level the data-releasing mechanism is able to provide for any arbitrary dataset(worst case guarantee), rather than directly measuring the mechanism’s output given a particular data input(one-time adhoc measurement). Our second contribution is to introduce a randomized algorithm which can satisfy this strong definition of privacy while still preserving good data utility.

We still adopt the same graph model, HRG, in this algorithm. The critical difference is that we impose randomness on the distribution from the model’s structure(i.e., the output of the original algorithm), instead of only enforcing randomness on the output itself.

As it is being pointed out, “Mathematically, anything yielding overly accurate answers to too many questions is non-private” [DP13]. In order to guarantee a strict sense of privacy, DP requires not only enforcing randomness on the answers but also restrain the number of queries being asked. One can quantify exactly the privacy loss in terms of the number of questions being answered, and in turn treat acceptable privacy loss as a budget that can be distributed to answer questions. However, with only limited access to the original data, it turns out to be very challenging to pick the right set of queries to effectively approximate the data’s properties. Furthermore, to guarantee good data utility, effective DP approaches also require the query’s sensitivity to be sufficiently low. In other words, the addition or removal of one arbitrary record should only incur limited change in the privacy-aware mechanism’s output distribution. Unfortunately, many existing approaches are not able to meet these challenges, i.e., they cannot provide reasonably good data utility guarantee after their data sanitization procedures.

Most existing DP schemes rely on the injection of Laplacian noise to add uncertainty to the query output, or more precisely, transform any pre-determined output to be a random sample from a statistical distribution. We, however, advocate a different approach that introduces uncertainty to queries directly. That is, we first use the HRG model to construct an output space, and then calibrate the underlying query distribution by sampling from the entire output space. Meanwhile, we make sure the series of sampled queries are independent of each other. Hence, the sensitivity of our scheme can be controlled to the magnitude of $\log n$, where n is the network size, as compared to $O(n)$ and $O(\sqrt{n})$ in state-to-art competing schemes [SZW+11; WW13; WWW13]. Intuitively, this indicates our scheme demands much less noise to be injected in perturbing the original data than other schemes.

From another prospective, as we draw random queries from a calibrated distribution, the set of sampled queries are unlikely to be the optimal for approximating the original data; however, we can still expect that, as long as the queries are good

enough, the resultant data utility should still be reasonably good. To further evaluate the effectiveness of our scheme, we also conduct empirical experiments on four real world datasets. Results show that the proposed method can still preserve good data utility even under stringent privacy requirements.

1.1.2 Collaborative access control

Next, we turn our attention to the individual user’s perspective and study an equally pressing problem. As mentioned above, besides the potential privacy loss caused by unrestrained collection and usage of OSN data, another major reason for unexpected privacy disclosure is due to user’s failure in managing the privacy settings to meet his/her privacy expectation. Ideally, one can always effectively limit the disclosure of information with sophisticated access control rules. However, OSNs today still lack tools to guide users to correctly manage their privacy settings. Hence, it is very important to develop practical tools that can relieve users from trivial maintenance of their privacy settings. To this end, the third contribution of this thesis is to develop such a tool for managing the access control policy in OSNs with ease.

In this work, we focus on the problem of collaborative access control. In today’s OSNs, it is common to see many online contents are shared and co-owned by multiple users. For example, Facebook allows a user to share his photos with others and tag the co-owners, i.e., friends who also appear in the photos. However, so far Facebook only provides very limited access control support where the photo publisher is the sole decision maker to restrict access. There is thus an urgent need to develop mechanisms for multiple owners of the shared content to collaboratively determine the access rights of other users, as well as to resolve the conflicts among co-owners with different privacy concerns. Many approaches to this question have been devised, but none of them consider one critical difference between OSNs and traditional scenarios, that is, rather than competing with each other and just wanting one’s own decision to be executed in traditional scenarios, OSN users may be affected by their peers’ concerns and adjust their decisions accordingly. As such, we approach the same collaborative access control problem from this particular perspective, integrating such *peer effects* into the strategy design to provide a more “considerate” collaborative access control tool.

Our solution is inspired by game theory. In this work, we formulate a game theory model to simulate an emotional mediation among multiple co-owners and integrate it into our framework named CAPE. Briefly, CAPE considers the intensity with which the co-owners are willing to pick up a choice (e.g. to release a photo to the public) and the extent to which they want their decisions to be affected by their peers' actions. Moreover, CAPE automatically yields the final actions for the co-owners as the mediation reaches equilibrium. It frees the co-owners from the mediation process after the initial setting, and meanwhile, offers a way to achieve more agreements among the co-owners. To materialize the whole idea, we also implement an app on a real OSN platform, Facebook. Details of the design and user interface will also be presented.

1.1.3 Thesis Organization

This thesis proceeds as follows. In Chapter 2, we will look at the background of network data releasing problems. We will review recent progress on defining privacy, as well as existing works for network data releasing that deploy different privacy definitions. In Chapter 3, we will present LORA, a randomization data perturbation method based on anonymization. Chapter 4 is then devoted to another mechanism that adopts a disparate privacy model – differential privacy. Next, we will introduce collaborative access control and motivate the problem in Chapter 5. Chapter 6 will then present our proposed peer-aware collaborative access control tool in details. We will conclude our work by summarizing our contributions and discussing directions for future work in Chapter 7.

The research in this thesis has been published and reported in various international conferences [XWT11; XCT14; XT12].

Chapter 2

Background and Related Works of OSN Data Publishing

In this chapter we review the background and related works on OSN data publishing. We give a brief history of privacy research by looking at how the academia started off to understand it, how the various academic disciplines have contributed to its understanding in recent years, and lastly, how our work fits into this discovery journey.

2.1 On Defining Information Privacy

Privacy, probably a bit surprising to see, is in fact a pretty modern concept. Western cultures have little formal discussion of information privacy in law until late 18th century [WB90]. The study of information privacy started off with the notion of anonymization, a definition aiming at removing *personally identifiable information* to prevent identity objects from being re-identified. The concept *personally identifiable information* (PII) now is frequently used in privacy laws to describe any information that can be used to uniquely identify an individual, such as names, social security numbers, IP addresses, etc. In particular, a set of several pieces of information that each of them is not PII by itself, can be combined to form a PII. In this case, it is called a *quasi-identifier* (QID).

In the study of privacy-preserving data publishing, it is commonly assumed an attacker who can use any methods or auxiliary tools to learn exact information of individual users. One type of notable attacks is called *linkage attack*, where the attacker can

re-identify individual users by joining different data resources(e.g., database, auxiliary background information) via QIDs. Apparently, under such attack, simply removing QIDs in each data source separately is inadequate to prevent re-identification. This is because combining multiple releases from different data sources can easily form new QIDs. To limit attackers' such ability to link to other information/data resources using QID and in turn thwart the risk of linkage attack, Sweeney proposed the notion of *k-anonymity* [Swe02]. *k-anonymity*, as well as other works in the same spirit such as *l-diversity* [MKG+07] and *t-closeness* [LLV07], are all based on the idea of hiding an individual in a crowd so that no individual's identity can be distinguished from the others in the crowd. We can categorize these works into the group that achieves anonymity by indistinguishability. In parallel to this group was another family of works, namely, anonymity by randomization. As suggested literally, this type of works usually randomly perturb the data source(e.g., add or delete records) to limit the attacker's confidence of certainty on the information he can obtain.

Comparing to randomization techniques, the main advantage of the former approach(*k-anonymity* [Swe02] and notions akin to this idea) is that it can provide a data-independent privacy guarantee. Hence comparatively, the former privacy model had attracted more attention and has been widely-adopted in many privacy-preserving data publishing works.

For decades, both academia and the society consider anonymization to be robust enough for effectively eliminating the privacy risk after each release of data. In other words, it is a "release-and-forgot" strategy [Pau09], a done deal after each release. The widespread adoption of anonymization makes it literally ubiquitous in our life. It has also been commonly accepted as the best practice to protect privacy both technically and legislatively. Big companies like Google also used to rely on anonymization techniques in practice to protect customers' privacy. Though acknowledged that "it is difficult to guarantee complete anonymization", they firmly believed that the anonymization techniques "will make it very unlikely for users to be identified" [Sog08].

However, a series of striking incidents challenged the presumption that anonymization can make re-identification difficult. In 2006, America Online(AOL) released 20 million search query logs to the public for research purpose. Even though the data is already suppressed and anonymized(i.e., identifiers such as names and IDs have been

removed), people soon found out that it was in fact quite easy to track a particular person with the released data [BJ06]. Two months right after this leakage, the famous Netflix Prize [Dem07] incident turned out to become the second warning that cast doubts on the effectiveness of anonymization techniques. Using the Netflix Prize dataset as an example, Narayanan demonstrated detailed de-anonymization techniques in [NS08] to show that k -anonymity failed to guarantee privacy. These revelations have shaken researchers' faith in anonymization as an effective mechanism for privacy protection. Prompted by these failures and acknowledging the critical defects of k -anonymity, researchers consequently proposed a series of improved privacy notions, e.g., l -diversity [MKG+07], t -closeness [LLV07]. Each was aimed at patching some flaws of the previous privacy notion based on anonymization, hoping to provide a stronger notion of privacy that can make re-identification difficult. However, as formally demonstrated in [NS08] and [Agg07], neither k -anonymity nor randomization methods can protect privacy on high-dimensional datasets [NS08; Agg07]. In fact, it is always possible (often also quite easy) to re-identify a person given enough auxiliary information or background knowledge. Attackers can always utilize cross-relations between data's attributes to trigger linkage attacks, rendering all anonymization-based strategies completely incapable to prevent re-identification.

Having identified and acknowledged the fatal defects of anonymity, differential privacy (DP) was proposed as a substitute to provide full protection against linkage attacks [DMN+06]. This definition was introduced in 2006 from the statistical disclosure community. The goal of DP is to form an adequate and principled definition that can quantify "privacy" in a rigorous sense under arbitrary attacks. To this end, differential privacy requires, no matter what auxiliary background knowledge that an attacker can have, the attacker will learn roughly the same information (the information disclosure is within a small multiplicative factor) no matter whether the individual participates in the database or not [DMN+06]. This worst case guarantee and clear semantic interpretation equip differential privacy to be a very strong and yet database-friendly privacy definition.

Mathematically speaking, DP requires any small changes in the input database should only result in small changes in the distribution of the output. As it turns out, DP is formalized within a mathematically rigorous framework. This lays a solid foun-

dation for DP and equips it to a useful formulation since many existing mathematical tools can be used to analyze and fulfill such definition.

The above apparent advantages, as well as its nice composition property, and a few known mechanisms found so far that achieve its formal requirement [DP13], leads differential privacy soon become an emerging de facto standard of information privacy.

2.2 On Practicing Privacy in Social Networks

With the increasing prevalence of social networks, the problem of privacy-preserving network data publishing has attracted substantial research interest. However, the nature of the complexity of social network data makes it much harder to apply any privacy models on it than on tabular data. Figure 2-1 depicts a timeline of the development in this research arena. It lists a few representative works on privacy notions and related privacy-preserving techniques in chronological order. It is easy to see that works on social networks clearly lag behind works on traditional tabular data (i.e., the same time when the privacy definitions were initially proposed). In this section, we will first review the early works that employed k -anonymity and randomization as the privacy model. We will also highlight the problems as applying anonymization on social networks. Lastly, we will turn to the recent development as applying differential privacy on the same network data-publishing problem.

2.2.1 Applying k -anonymity on social networks

Backstrom et al. [BDK07] point out that naive anonymization practice such as just removing user ids or names in social graphs poses serious threaten to user privacy. Recall that anonymization requires all PII's to be sanitized. However, in networks, such "data-fingerprint" PII can turn out to be many different forms. That is, the attacker can uniquely identify an individual in graphs via many graph patterns, such as node's degree, subgraph, hub, node attribute and neighborhood structure. To protect against the attacks on these PII's, the majority of work based on k -anonymity had defined various ad-hoc definitions, each assumes a particular type of adversarial knowledge. For example, Liu and Terzi [LT08] propose k -degree anonymity that requires

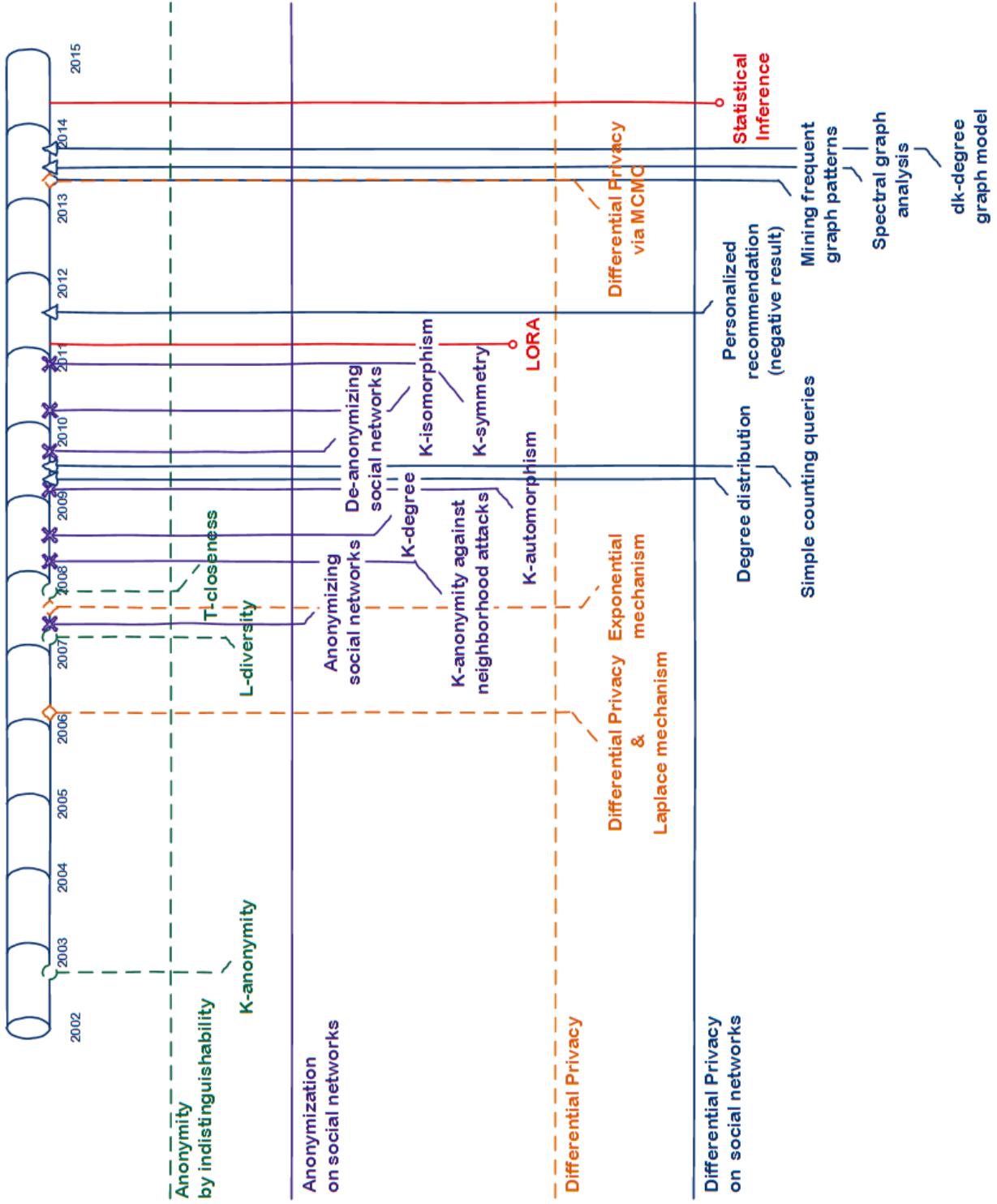


Figure 2-1: Timeline of Selected Works on Privacy-preserving Data Publishing

that, for every node v in the network, there exist at least $k - 1$ other nodes with the same degree as v . Zhou and Pei [ZP08] demand that each node should have the same 1-neighborhood structure as at least $k - 1$ nodes. Zou et al. [ZCO09] propose k -automorphism, which enforces $k - 1$ automorphic functions in the anonymized data. Cheng et al. [CFL10] introduce the notion of k -isomorphism, which requires an input graph to be transformed into k disjoint isomorphic subgraphs. Yuan et al. [YCY10] allow the user to customize their privacy protection needs via defining k -anonymity on different strength of the attacker’s background knowledge.

In a nutshell, the above methods all adopt the same paradigm to achieve anonymity: using deterministic methods to alter the network structure in order to satisfy some types of structure uniformity. In parallel to this family of works, k -anonymity can also be achieved via node generalization and suppression [CT08; CSY+08; BCK+09; HMJ+08]. For example, Hay et al. [HMJ+08] investigate applying generalization and suppression techniques on nodes to achieve k -anonymity.

Broadly, the goal of all these works is to scrub the original data to remove a particular type of “data-fingerprint” in the social graphs, while at the same time restrain the amount of modification(i.e., information loss) upon the data to be as little as possible. However, subjected to the drawbacks of k -anonymity, all k -anonymous network sanitation techniques are vulnerable to attackers with stronger background knowledge than assumed. In hindsight, the line of these works seems to be trapped in a cycle of “identify–anonymize–re-identify–anonymize again”. There is to date still no satisfactory definition that offers a general concept of k -anonymity on social networks precisely.

2.2.2 Applying anonymity by randomization on social networks

Another line of works considers randomization to be the privacy model. Rather than protecting the nodes by constructing structural uniformity base on k -anonymity mentioned above, most works in this family directly perturb the links (a.k.a. randomly add/delete edges). The direct effect of randomization is to limit the attacker’s confidence as he attempts to infer the existence of true edges in the network. The node’s identity in turn can also be effectively protected with high probabilities, since the formation of most PII’s often rely on some structure patterns consisting of the

links. Hay et al. explore this problem in [HMJ+07] by introducing an anonymization framework based on edge perturbation. Empirical experiments in this report demonstrate that such strategy can substantially reduce the risk of privacy breach. Ying et al. [YW08] further explore the same problem by considering graph's spectra as an indicator to navigate the choices of links to add/delete during the perturbation. In [YW09] and [HGP09], Ying and Hanhijarvi consider generating synthetic graphs with Metropolis-Hasting algorithms. Essentially, both works extract statistical summaries of the original graph (e.g., degree distribution and average clustering coefficient and characteristic path length), and then use Metropolis-Hasting method to sample the set of graphs with same parameters as the original graph.

Compared with k -anonymous methods, randomization have a few very attractive advantages for network anonymization problems. First of all, it is not subjected to a particular type of attack, which is the main limitation of k -anonymity methods. Secondly, the flexible nature of randomization allows a great amount of perturbation on the real-world network data(which is usually large and sparse) without significant deteriorating the network structure. Even though in literature, some empirical evaluation on moderate-sized datasets in [WYW10] suggests the network's topological features "will be significantly lost in the randomized graph when a medium or large perturbation is applied". Ying et al. [YPW+09] also compared the randomized edge perturbation method to k -degree anonymous method proposed in [LT08] using three moderate-sized datasets, and reach the conclusion that k -degree method preserves better network properties. However, we should stress that such randomization approaches' privacy-preserving ability is data-dependent. The two above works both demonstrate empirical evaluation only on moderate-sized datasets(*polblogs* with 1,222 nodes, 16714 edges; *polbooks* with 105 nodes, 441 edges; *Enron* with 151 nodes, 869 edges). Bonchi et al. [BGT11] argue that previous works in fact underestimate randomization's competence in solving privacy-preserving problems. They demonstrate on real-world datasets randomization strategy can yield meaningful privacy protection while still preserving good network properties. They also point out that *posterior* belief probability, the metric previously used to assess randomization techniques' privacy-preserving level in many works, is rather a local measure of privacy level. They advocate to use entropy as a more global-sense measure to quantify randomiza-

tion’s ability in preserving privacy. Moreover, they further extend their work in [BGT14] to show the detailed analysis of how to quantify random perturbation’s resilience to attacks.

Our first work can also be categorized into this line of works. Specifically, we adopt a *hierarchical random graph*(HRG) graph model [BGT11] to randomly perturb the links in the networks. We show that the best-fitting HRG model carefully capture all “link equivalent class”, in which all links play similar roles in topology globally and locally. The advantage of such a method is that it can tailor the network with regard to the network’s own structure while allowing large amount of edge perturbation on it. Besides, inspired by [BGT11], we formulate “link entropy”, the counterpart of “identity entropy” in [BGT11], to quantify the link privacy of our methods from the perspective of information theory.

For more detailed account on applying anonymity on network data-publishing, we refer interested readers to a few nice surveys [FWC+10; AMP10; ZPL08] and a tutorial in [HLM+11].

2.2.3 Applying differential privacy on social networks

Recently, differential privacy has been widely investigated in privacy-aware data mining and data-publishing communities. Its success stems from its rigorous privacy guarantee, as well as its nice formulation as an interactive mechanism, where the analyst can only query the database and collect the answer without full access to the raw data. This particularly facilitates the development of applying DP to gain certain statistical results via posing queries. Specifically in networks, a line of works along this direction aims to release certain differentially private data mining results, such as degree distributions, subgraph counts and frequent graph patterns [HLM+09; KRS+11; HR12; SY13]. Hay et al. [HLM+09] make use of the constrained inference technique to release a private estimate of a network’s degree distribution. Karwa et al. [KRS+11] approximate answers to different subgraph counting queries based on *local sensitivity* and *smooth sensitivity*, which achieves weaker privacy guarantee. Hardt and Roth [HR12] give an efficient algorithm for finding a low rank approximation of a matrix. Shen and Yu [SY13] consider the problem of frequent graph pattern mining by proposing a MCMC sampling based algorithm.

However, the problem we confront, the task of full release of network data, actually falls into another direction of problems. Our goal is to employ DP in the task of synthetic data generation. This essentially seeks to approximate all functions that a network possesses. Clearly, publishing the entire networks is much more challenging than publishing just certain network statistics or data mining results. The main obstacle to publish the entire graph can easily incur a large global sensitivity. Note that the sensitivity in the problem setting of [SY13] is only 1. In contrast, existing works dealing with graph releasing problems often have much larger sensitivities. Compared with these state-of-the-art competitors, our key technical contribution in our second work is to achieve a much smaller sensitivity in releasing a graph (i.e., $O(\log n)$ as opposed to $O(n)$ and $O(\sqrt{n})$ in [SZW+11; WW13; WWW13]).

Inspired by [SY13], our second work also utilize MCMC sampling strategy to achieve differential privacy. We still use HRG as the graph model in this work. But, instead of directly enforcing random perturbation on MCMC's output (as in our first work), our second work carefully calibrates the underlying distribution of MCMC to meet differential privacy's requirements. By sampling the entire HRG space, the algorithm can reap both differential privacy and good data utility simultaneously.

It worths pointing out that even though based on the same graph model, HRG, our first and second work instantiate the concept of privacy with two disparate paradigms. The first work looks at the best-fitting HRG model itself and look for the room to perturb the data while preserving the original network topology. In this case, the privacy guarantee is data-dependent, relying on the network's own structure. Conversely, in the second work, the privacy guarantee is strictly fulfilled by differential privacy. We aim to treat graph itself as statistical data, that is, the original network can be considered as a random sample drawn from an underlying distribution. By carefully inferring back such distribution and calibrating it with regard to DP, we can harvest uncertainty and privacy via sampling procedure. In some sense, the second method is a reminiscent of classical statistical inference problems.

Chapter 3

LORA: Link Obfuscation by RAndomization in Social Networks

3.1 Introduction

Information on social networks are invaluable assets for exploratory data analysis in a wide range of real-life applications. For instance, the connections in OSNs(e.g., Facebook and Twitter) are studied by sociologists to understand human social relationships; co-author networks are explored to analyze the degree and patterns of collaboration between researchers; voting and election networks are used to expose different views in the community; trust networks like Epinions are great resources for personalized recommendations. However, many of such networks contain highly sensitive personal information, such as social contacts, personal opinions and private communication records. To respect the privacy of individual participants in social networks, network data cannot be released for public access and scientific studies without proper “sanitization”.

In this work, we consider simple graphs to represent network data, where the nodes capture the entities and the edges reflect the relationships between the entities. For example, in social networks such as Facebook (facebook.com), a graph captures the friendships (edges) between individuals (nodes). Our goal is to preserve personal privacy when releasing such graphs.

While there has been numerous attempts along this line of works, these methods are still vulnerable to various types of attacks [LT08; ZP08; BDK07; HMJ+08]. Back-

strom et al. [BDK07] show that, with very limited background knowledge, a large number of nodes can be easily re-identified even after sanitizing the node’s identity information such as social ID and name. More recently, Liu et al. [LT08] report that the degree of a node can be used as a quasi-identifier to re-identify the node’s identity in the graph. Zhou et al. also claim that local subgraph knowledge such as a node’s neighborhood can be easily retrieved by attackers. By matching the structure of the victim node’s subgraph, attackers can trace and find the victim node [ZP08]. Hay et al. [HMJ+08] also point out that hubs, as the fingerprints of graphs, are often uniquely identifiable. In fact, the popularity of social networks in recent years and the availability of powerful web crawling techniques have made accessing personal information much easier to achieve. Therefore, it is almost impossible to foresee an attacker’s background knowledge in advance. Meanwhile, it is also unrealistic to make any assumptions on the constraints of an attacker’s ability to collect such knowledge. As such, it is challenging to preserve privacy on graphs. This has prompted researchers to develop robust network/graph data protection techniques.

Existing works on preserving privacy of graphs fall into two main theoretical privacy models: k -anonymity-based model [ZCO09; WXW+10; CFL10; LT08] and randomization model [HMJ+08; BGT11]. Under the former privacy model, a source graph is manipulated so that it has at least k corresponding entities satisfying a same type of structural knowledge. However, these methods are designed to be robust to certain specific attacks. For example, k -degree [LT08] and k -automorphism [ZCO09] anonymization schemes are specially designed to protect the privacy of node degrees. Moreover, these works typically assume the attackers’ background knowledge is limited. In addition, graph modification is often restricted as the released graphs need to respect some symmetric properties in order for k candidates to share certain properties in the graph.

On the other hand, in randomization models [BGT11; YW08; YW09; HGP09], the released graph is picked from a set of graphs generated from a random perturbation of the source graph (through edge addition, deletion, swap or flip). Such an approach offers more freedom in “shaping” the released graph, i.e., no additional properties are intentionally injected. More importantly, an attacker’s background knowledge would become less reliable because of the random process. For example, by allowing

random insertion and deletion of edges, an attacker is no longer 100% certain of an edge’s existence. Moreover, randomization techniques are typically designed to be independent of any specific attacks, and hence are robust to a wider range of attacks. However, uncontrolled random perturbation means the space of the distribution from which the released graph is picked is effectively “unbounded”, making it difficult to preserve the source graph’s structure. For example, if we allow only edge deletion, since edges are arbitrarily selected for deletion, important ties in a graph, such as bridge edges, may be eliminated resulting in a partitioned graph.

In this work, we advocate and focus on randomization techniques. Our goal is to ensure that the released graph is privacy preserving, and yet useful for a wide range of applications. In particular, for the latter, the released graph should be “similar” to the source graph in terms of most properties (e.g., degree distribution, shortest path length and influential nodes). This raises three questions:

1. How to randomize a source graph so that the resultant released graph is still similar to it?
2. How to provide a measurement of shared information between the source and released graphs, to indicate the utility of the released graph? Conversely, the measurement reflects the information loss due to randomization.
3. How to quantify the effectiveness of the randomized technique (and randomized graph) with regard to privacy preservation? In other words, what is an appropriate measurable definition of privacy on graph?

From existing works, we can see much effort to address the first question above. In [YW08], the proposed approach restrains the changes in the random graphs’ spectra to provide rough bounds of the random graph distribution. Another approach adopts the Metropolis-Hastings algorithm (specifically, the Markov Chain Monte Carlo method) to sample graphs with feature constraints [HGP09; YW09]. This approach can preserve several graph statistical summaries, such as degree distribution, average clustering coefficient and average path length. However, since many statistical summaries typically provide descriptions of a graph from different perspectives, but do not directly determine the graph structure, it is hard to quantify information lost since other graph features are not intentionally preserved. It is also not easy to evaluate its

effectiveness with regard to privacy preservation. In these works, the popular privacy measurement adopted merely relies on the different numbers of edges between the two graphs [YW09].

In this chapter, we propose a randomization scheme, LORA (Link Obfuscation by RAndomization), to generate a synthetic graph (from a source graph) that preserves the *link* (i.e., the extent of two node’s relationship) while blurring the existence of an edge. In our context, *link* refers to the relation between two nodes. It is a virtual connection relationship, and is not necessarily a real edge that physically exists in the graph. We use the concept *link probability* as a quantity to measure the strength of *link*.

Next, we explain how LORA addresses the three questions that we raised. Firstly, we adopt the hierarchical random graph (HRG) model [CMN07; CMN08] to estimate each link probability in the source graph. The HRG model is a generic model that can capture assorted statistical properties of graphs. Based on the HRG model, we can randomly generate graphs that are similar to the source graph with regard to statistical properties (i.e., dealing with the first challenge). Next, by reconstructing statistically similar graphs that preserve the source graph’s HRG structure, we can select one to be released. In the ideal scenario, the released graph and source graph would share exactly the same HRG structure (i.e., addressing the second challenge).

Third, to investigate how our method can preserve link privacy and how to quantify its strength, we introduce the notion of *link entropy*. Entropy has been widely used to measure the uncertainty of random variables in information theory. We will show that entropy is also appropriate in our scheme in terms of clarification and simplicity, compared to posterior belief that is used in previous works. Instead of analysing privacy with node’s entropy [BGT11], we define entropy based on links to theoretically quantify the effectiveness (regarding privacy preservation) of our randomization scheme. As an attempt to address the third challenge, we will show how to derive the entropy for each individual link and then the composition of entropy of a set of links. We specifically define the notion of entropy of a node’s egocentric network, which is an entropy ensemble and quantifies our scheme’s privacy-preserving strength towards egocentric subgraphs. We will show how entropy quantifies an attacker’s uncertainty accurately and clearly towards an egocentric network.

The rest of this work is organized as follows. In Section 3.2, we provide some preliminaries. Section 3.3 gives an overview of our proposed LORA, and Section 3.4 presents the technical details of LORA. In Section 3.5, we analyze the privacy of our proposed LORA. Section 3.6 presents results of experimental studies. Finally, we conclude this work in Section 3.7.

3.2 Preliminaries

3.2.1 Graph Notation

In this study, we follow the convention to model a network as a simple undirected graph $G = (V, E)$. V is the set of vertices and $E \subseteq V \times V$ is the set of edges. Let $|V| = n$ and $|E| = m$.

A mathematical representation of G is the adjacency matrix of G . We denote it with $A \in \{0, 1\}^{n \times n}$. $A_{ij} = 1$ if there is an edge between vertices i and j in G and $A_{ij} = 0$, otherwise. Moreover, we use $\tilde{G}(\tilde{n}, \tilde{m}) = (\tilde{V}, \tilde{E})$ to denote the released graph reconstructed by randomization.

3.2.2 Hierarchical Random Graph and its Dendrogram Representation

A graph often exhibits a hierarchical organization. Vertices can be clustered into subgraphs, each of which can be further subdivided into smaller subgraphs, and so forth over multiple scales. The hierarchical random graph (HRG) model is a tool to explicitly describe such hierarchical organization at all scales for a graph. According to Clauset’s experiments [CMN08], the graphs “resampled” with HRG can match the statistical properties of the source graphs closely, including degree distributions, clustering coefficients, and distributions of shortest path lengths.

The hierarchical structure of G in an HRG is captured by a *dendrogram* T , which is a rooted binary tree with n leaf nodes corresponding to the n vertices of G . Each internal node r of T is associated with a probability p_r . For any two vertices i, j in G , their probability of being connected $p_{ij} = p_r$, where r is their lowest common ancestor in T . Formally, an HRG is defined by a pair $(T, \{p_r\})$.

Let L_r and R_r be the left and right subtrees of r respectively. n_{L_r} and n_{R_r} are the numbers of leaves in L_r and R_r . Let e_r be the number of edges in G whose endpoints are leaves of each of the two subtrees of r in T . The *likelihood* of an HRG for a given graph G can be calculated, by Bayes' theorem, as follows:

$$\mathcal{L}(T, \{p_r\}) = \prod_{r \in T} p_r^{e_r} (1 - p_r)^{n_{L_r} n_{R_r} - e_r} \quad (3.1)$$

For a fixed dendrogram T , the maximum likelihood estimator of p_r is $\frac{e_r}{n_{L_r} \cdot n_{R_r}}$. It represents the fraction of potential edges between the leaves of L_r and R_r that actually exist in G . In our scheme, we work with the logarithm of the likelihood :

$$\log \mathcal{L}(T, \{p_r\}) = - \sum_{r \in T} n_{L_r} n_{R_r} h(p_r) \quad (3.2)$$

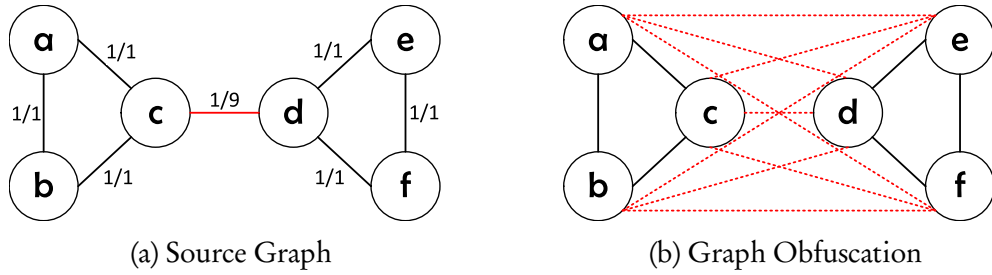
where

$$h(p_r) = -p_r \log p_r - (1 - p_r) \log(1 - p_r) \quad (3.3)$$

is the Gibbs-Shannon entropy function.

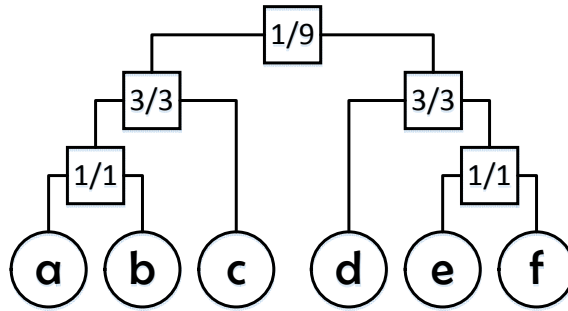
Essentially, the likelihood of a dendrogram measures how plausible this HRG is to represent a graph. A dendrogram paired with a higher likelihood is a better representation of the network's structure than those with lower likelihoods. We denote $\log \mathcal{L}(T, \{p_r\})$ by $\log \mathcal{L}(T)$ from now on when no confusion arises.

The best-fitting HRG of an original graph can be obtained using the Markov Chain Monte Carlo method (MCMC). In practice, most real world networks will have many plausible hierarchical representations of roughly equal likelihood, which may slightly differ in arrangement of tree's branches. We sample dendrograms at regular intervals and calculate the mean probability p_{ij} for each pair of vertices (i, j) . In our analysis, we assume the dendrogram derived by MCMC is always the ideal one that fits the source data best. For instance, we assume Figure 3-1c is Figure 3-1a's best-fitting dendrogram. From Figure 3-1c, we note that all p_{ij} can be quantified with $\frac{e_r}{n_{L_r} \cdot n_{R_r}}$ as shown in the probability matrix in Table 3-1d.



(a) Source Graph

(b) Graph Obfuscation



(c) Best-fitting Dendrogram

	v_a	v_b	v_c	v_d	v_e	v_f
v_a	1	1	1	1/9	1/9	1/9
v_b	1	1	1	1/9	1/9	1/9
v_c	1	1	1	1/9	1/9	1/9
v_d	1/9	1/9	1/9	1	1	1
v_e	1/9	1/9	1/9	1	1	1
v_f	1/9	1/9	1/9	1	1	1

(d) Link Probability Matrix

	v_a	v_b	v_c	v_d	v_e	v_f
v_a	0	0	0	0.50	0.50	0.50
v_b	0	0	0	0.50	0.50	0.50
v_c	0	0	0	0.50	0.50	0.50
v_d	0.50	0.50	0.50	0	0	0
v_e	0.50	0.50	0.50	0	0	0
v_f	0.50	0.50	0.50	0	0	0

(e) Link Entropy Matrix

Figure 3-1: An example of HRG model in [CMN08; CMN07].

Example 3.1. Figure 3-1c is a best-fitting dendrogram representation of the graph in Figure 3-1a. At the top scale of dendrogram in Figure 3-1c, vertices in graph G are divided into two groups from the root r in dendrogram T , corresponding to the leaf set $\{a,b,c\}$ in the left subtree of T and leaf set $\{d,e,f\}$ in the right subtree, respectively. Each group has 3 leaf nodes, so $n_{Lr} = n_{Rr} = 3$. Since only one edge (c,d) exists in the real graph, e_r should be one. And the probability p_r of connections between two groups can be estimated as $p_r = \frac{e_r}{n_{Lr} \cdot n_{Rr}} = \frac{1}{9}$. ■

3.2.3 Entropy

Entropy measures the uncertainty regarding the value of one random variable in information theory. The less probable the outcome of one random variable X is, the greater its entropy is.

A random variable X has a probability p to render an outcome x , and a probability $1 - p$ to generate another alternative outcome x' . The uncertainty of an outcome of this random variable X is defined as a binary entropy function,

$$H(X) = -p \log_2 p - (1 - p) \log_2 (1 - p) \quad (3.4)$$

with the convention that $0 \times \log 0 = 0$.

An ensemble random variable X , where the outcome x is the value of X , can take on one of a set of possible values, $\mathcal{C}_X = \{c_1, c_2, \dots, c_K\}$. \mathcal{C}_X has probabilities $\mathcal{P}_X = \{p_1, p_2, \dots, p_K\}$. The entropy of the ensemble variable X is,

$$H(X) = H(p_1, p_2, \dots, p_K) = -\sum_{k=1}^K p_k \log_2 p_k, \quad (3.5)$$

Entropy has additive properties for independent variables. That is, if variable X and Y are independent, the entropy of the outcome (x, y) satisfies,

$$H(X, Y) = H(X) + H(Y) \quad (3.6)$$

In addition, $H(X) \geq 0$ with equality if and only if p_k equals to 0 or 1 for each k .

3.3 LORA: The Big Picture

Our proposed LORA framework consists of two main steps: (1) Find an HRG model that fits the source graph best; (2) Based on the best-fitting HRG, reconstruct a new graph by random link sampling. Algorithm 3.1 outlines the framework of LORA.

We firstly introduce two critical concepts: *link* and *link probability*. In our context, the term *link* refers to the relation between two nodes. The term *link probability* is a quantity to measure the strength of *link*. These two concepts are appropriate to depict such a scenario: A pair of nodes, although not directly connected in the source graph, may still have a weak relation (*link*) if the two share many common neighbours. Due to these common neighbours, a promising connection may appear in the future. For example, in social networks, friends of friends would more likely become friends soon. To distinguish, we use the term “edge” if there is a direct connection between two nodes in a graph.

The role of each link differs in its impact on topology. For instance, in Figure 3-1a, links (c, d) and (a, f) exhibit the same topological effect, i.e., they are exchangeable. Thus, we can replace edge (c, d) in Figure 3-1a with link (a, f) without sacrificing any topological structure (see Figure 3-2). Moreover, the role of vertex is fully determined by the links incident upon the vertex. As Figure 3-1a shows, vertices c and d have the same roles, and so are vertices a and b .

In order to estimate such *link probabilities* in the source graph, we adopt the hierarchical random graph (HRG) model in LORA. More specifically, we use the Markov Chain Monte Carlo (MCMC) method to find the HRG model that best fits the source graph. We choose HRG as our model because it is a generic model, which describes a graph’s structure in detail, including all the probabilities of a connection between any two vertices in the graph. Here we extend the concept of this probability to be *link probability* in order to quantify our “link” notion. In addition, in [CMN08], Clauset et al. claim that the HRG model can capture assorted statistical properties of a graph. It is also shown that hierarchy is a central organizing principle of networks [CMN08]. In contrast to simple clustering, HRG describes the vertices’ organization at all scales in a network. The differences between graphs generated with one specific HRG would be limited. Hence, HRG can effectively bound the distribution of regenerated random graphs.

Algorithm 3.1: The LORA Framework

Data: A simple source Graph $G(V, E)$

Result: A reconstructed random Graph $\tilde{G}(\tilde{V}, \tilde{E})$ for release, where $\tilde{V} \subseteq V$

```
1 Dendrogram  $T \leftarrow \text{fitHRG}(G)$ 
2 foreach non-leaf node  $r$  in  $T$  do
3    $V_{\text{left}}(r) \leftarrow \text{findLeafVertices}(\text{left subtree of } r)$ 
4    $V_{\text{right}}(r) \leftarrow \text{findLeafVertices}(\text{right subtree of } r)$ 
5   link equivalent class  $L(r) \leftarrow V_{\text{left}}(r) \times V_{\text{right}}(r)$ 
6    $e_r \leftarrow$  the number of observed edges  $\in V_{\text{left}}(r) \times V_{\text{right}}(r)$  in  $G$ 
7   randomly pick up  $e_r$  links in current equivalent class  $L(r)$  to be new edges in  $\tilde{G}$ 
8 end
```

Now, we begin to describe the two steps in LORA. At the first step of LORA, we determine the best-fitting HRG model of the source graph by running MCMC sampling algorithm until equilibrium and represent it as a dendrogram tree T (See Algorithm 3.1, line 1). Leaf nodes in T correspond to vertices in the graph. Each non-leaf internal node is associated with two communities (i.e. two sets of leaf nodes) induced by its left and right subtrees (lines 3, 4). Ideally, links across these two communities are viewed as approximately equivalent and exchangeable relationships in terms of the inter-community association strength. We denote such a group of equivalent links as one link equivalent class (line 5).

Secondly, in the reconstruction step (lines 6, 7), we replace true edges in the source graph with their equivalent links in link equivalent classes. In order to maintain the same inter-community association strength, we randomly pick the same number of links in the link equivalent class to substitute the true edges observed in the source graph. We then set the chosen links to be the new edges in the released graph (line 7). We should stress that link obfuscation also comes simultaneously from such random process. In the privacy analysis part, we would introduce the concept of “*link entropy*” to assess the degree of privacy brought by link obfuscation.

3.4 Link Obfuscation by Randomization with HRG

3.4.1 Link Equivalence Class

Given a network $G(n, m) = (V, E)$ and its best-fitting HRG dendrogram tree T , links bridging the nodes in the left and right subtrees are in the same equivalence class with respect to their topological roles. Consider the graph G in Figure 3-1a and its best-fitting dendrogram T in Figure 3-1c. At the top level of T , the dendrogram divides into two subtrees, which induces two separate leaf sets - left subtree leaf set $\{a, b, c\}$ and right subtree leaf set $\{d, e, f\}$. All the possible cross links bridging these two leaf sets consist of one link equivalence class. In this case, as shown by the dash lines in Figure 3-1b, links $(a, d), (b, d), (c, d), (a, e), (b, e), (c, e), (a, f), (b, f), (c, f)$ are 9 pairs of links in one equivalence class. Let n_{Lr} and n_{Rr} be the sizes of the left and right leaf sets respectively (in our example, $n_{Lr} = 3, n_{Rr} = 3$). Let e_r be the number of edges in the source graph G linking the two sets (in Figure 3-1a, there is only one edge (c, d) , so $e_r = 1$). We can then estimate the link probability of this link equivalence class as $e_r / (n_{Lr} \cdot n_{Rr}) = 1 / (3 \cdot 3) = 1/9$. This probability indicates the connection strength between the nodes in the two leaf groups. As such, we are now ready to obfuscate the existence of connections of nodes by turning real edges into virtual probabilistic links.

Note that, ideally, if links in one equivalent class share exactly the same topological roles, the new generated graph should also share exactly the same dendrogram as the source graph. However, this is not usually the case. Very often, the equivalent link class derived through HRG is a group of approximately topological similar links. In this work's analysis, we assume the released graph shares exactly the same best-fitting (i.e., global optimal) dendrogram of the source graph. Note that, from a privacy's perspective, it is apparent that, if the best-fitting dendrograms of the source and released graphs are not the same, it would be even much harder to infer the source graph from the released graph. Therefore, this assumption is biased against LORA in terms of its privacy strength. Since the statistical relationship of each link in the graphs are fully preserved in this ideal scenario, such released graphs would share the same inherent statistical properties with the source graph. Essentially, all information that the released and source graphs share is the dendrogram T in the ideal scenario.

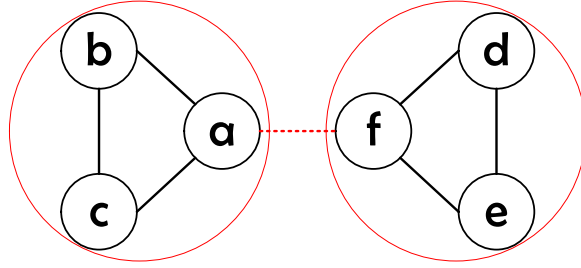


Figure 3-2: Perturbed Graph & Node Generalization

3.4.2 Link Replacement

Now we explain the link replacement procedure with an HRG during graph reconstruction. We reconstruct a random graph by a series of link replacement procedures, where each inner node in HRG corresponds to one link replacement procedure. Consider one inner node r in dendrogram T . There are e_r real edges in G bridging the two leaf node sets in the left and right subtrees in T . In order to maintain the same connection strength between the two leaf node sets in the released graph \tilde{G} , we randomly pick e_r links in an inner node r 's link equivalent class to replace the e_r real edges. The whole reconstruction process is done through $n - 1$ independent link replacement procedures, corresponding to $n - 1$ inner nodes in HRG. Referring back to our running example, let us consider the root node in the dendrogram. Since there is only one real edge (c, d) , we need to find a link to replace it. Figure 3-2 shows the released graph after link (a, f) replaces edge (c, d) .

3.4.3 Hide Weak Ties & Retain Strong Ties

Essentially, a best-fitting HRG tends to exchange weak ties (yet true edges) in the source graph with links that are not connected yet. See the instance of graph in Figure 3-1a and its perturbed graph in Figure 3-2. By “weak ties”, we mean edges that are bridges to link two strong-connected components in graphs. The edge between c and d in Figure 3-1a shows such a case. As real world graphs are typically sparse, weak ties are not uncommon. In fact, they are important channels between many clustered groups and hold important roles in shaping the entire graph structure. In pure random edge deletion schemes, such weak ties may be removed, which will severely undermine the source graph's structure. However, under our scheme LORA, link obfuscation is employed to substitute such weak ties with “fake” (non-existent) ties within the same

equivalence class. In this way, large amount of changes can be operated on the graph while preserving the skeleton of the source graph as well as the clique-like components.

Using Figure 3-2 as illustration, we associate (small) link probabilities with weak links, that is, links between leaf node sets $\{a, b, c\}$ and $\{d, e, f\}$, which give much freedom to perturb the source graph to obfuscate links in the released graph. In this case, leaf nodes $\{a, b, c\}$, which are rooted in the same inner node, have exactly the same link relationship towards all the other nodes. Therefore, they are interchangeable. In Figure 3-2, the skeleton (the dashed line and dashed circles) of the perturbed graph (of the graph in Figure 3-1a) remains the same as the source graph. For complete components, namely, clique, link probability obfuscation usually would preserve them fully since they link with each other closely. This makes sense since cliques are obvious features in graphs, one cannot perturb one complete graph without sacrificing its property of complete graph. To improve the privacy of nodes in a clique, an alternative way is to coalesce cliques into one super node. That is, in our context, to generalize all the leaf nodes induced from one subtree in the dendrogram to one supernode. Clearly, with a best-fitting dendrogram, it is very easy to identify effective ways to coalesce subcomponents of graph with minimal disruptions to the remaining structure.

3.5 Privacy Analysis

In this work, we use entropy as the privacy criterion to quantify the strength of link obfuscation method in preserving link privacy.

As an analogy of binary entropy function, each link A_{ij} with link probability p_{ij} has binary link entropy

$$H(A_{ij}) = -p_{ij} \log_2 p_{ij} - (1 - p_{ij}) \log_2 (1 - p_{ij}) \quad (3.7)$$

Link entropy quantifies the degree of uncertainty of the existence of a edge, i.e., whether nodes i and j are connected or not in source graph. A larger entropy value indicates better privacy. For example, the table in Figure 3-1e is a matrix consisting of all link entropies, which is derived by the probability matrix in Figure 3-1d for the graph in Figure 3-1a. As shown in Figure 3-1e, links between node set $\{a, b, c\}$ and set

$\{d, e, f\}$ have entropy 0.50, indicating the uncertainty level of the true state of the links measured in entropy.

3.5.1 The Joint Link Entropy

It is not uncommon for attackers to attempt to identify a set of links, e.g. checking the egocentric network of one node (i.e., all edges incident to that node), searching for subgraphs and so on. To this end, we now formalize the joint link entropy to quantify the degree of uncertainty in these scenarios.

Joint entropy of dependent links

For the links associated with the same inner node r in a dendrogram T , they are dependent (or relevant) random variables. Consider observing the outcome of K ($K \leq n_{Lr} \cdot n_{Rr}$) dependent links whose endpoints are rooted at the same lowest common ancestor r in T . We use a joint ensemble variable $X_r = A_{i_1 j_1}^r A_{i_2 j_2}^r \dots A_{i_K j_K}^r$ to represent the ensemble of such K link variables under observation. The ensemble variable X_r can take on one of a set of possible ensemble outcomes, x_r^s , which consists of the outcome of each link variable A_{ij}^r . Here $A_{ij}^r = 1$ denotes link A_{ij}^r is chosen during link replacement; otherwise, for non-chosen links, $A_{ij}^r = 0$. Each specific outcome x_r^s has a probability p_s . In the context of link replacement, p_s refers to the possibility one specific outcome x_r^s of the K relevant links (chosen or unchosen) appears after link replacement. The link ensemble X_r has a joint probability distribution $\mathcal{P}_{X_r} = \{p_{s_1}, p_{s_2}, p_{s_3}, \dots\}$ over all possible outcomes. We use s to denote the specific values taken by x_r^s . s is essentially a sequence consisting of 0 or 1. We use \mathbb{S} to denote the set of all possible such sequences that the outcomes of X_r can have.

As one example, we consider the outcome values(i.e. the possible outcomes) of links $X_r = a_{ad}a_{bd}a_{cd}$ in Figure 3-1a. X_r can take on one of 4 ordered sequence outcomes, that is, no link selected from $\{A_{ad}, A_{bd}, A_{cd}\}$ (outcome “000”); A_{cd} selected(outcome “001”); A_{bd} selected(outcome “010”) and A_{ad} selected(outcome “100”). Consider the calculation of $p_s(000)$, the probability that the outcome value of link set $\{A_{ad}A_{bd}A_{cd}\}$ is 000 after link replacement. First of all, after the link replacement regarding inner node r , there are $\binom{n_{Lr} \cdot n_{Rr}}{e_r}$ types of outcomes for the whole link equivalent class. Among all the outcomes, we count the number of outcomes where link $\{A_{ad}A_{bd}A_{cd}\}$

is 000. Sequence 000 indicates that in the observed link set $\{A_{ad}, A_{bd}, A_{cd}\}$, none of links is chosen. Let l denote the number of links chosen from the K relevant links. Hence, in sequence 000, $l = 0$. The observed link set size K is 3 here. In order to replace 1 ($e_r = 1$) original edge in G , another 1 (i.e., $e_r - l = 1$) link needs to be drawn from the rest links in the inner node r 's link equivalence class ($n_{Lr} \cdot n_{Rr} - K$ links). There are $\binom{n_{Lr} \cdot n_{Rr} - K}{e_r - l}$ types of ensemble outcomes for drawing $e_r - l$ links from the rest links. Hence $p_s(000)$ is $\binom{3 \cdot 3 - 3}{1 - 0} / \binom{3 \cdot 3}{1} = 6/9$. In the following, we give the generalized formula of p_s :

$$p_s = \frac{\binom{n_{Lr} \cdot n_{Rr} - K}{e_r - l}}{\binom{n_{Lr} \cdot n_{Rr}}{e_r}} \quad (3.8)$$

where l is the number of links drawn from the observed K relevant links.

The joint entropy of dependent links is defined as,

$$\begin{aligned} H(X_r) &= H(A_{i_1 j_1}^r A_{i_2 j_2}^r \dots A_{i_K j_K}^r) \\ &= H(p_{s_1}, p_{s_2}, p_{s_3}, \dots) = - \sum_{s \in \mathbb{S}} p_s \log_2 p_s \end{aligned} \quad (3.9)$$

which measures the degree of attacker's uncertainty regarding a set of dependent links.

As illustration, we again consider the possible outcomes of links $X_r = A_{ad} A_{bd} A_{cd}$ in Figure 3-1a. The space of X_r 's possible worlds consists of 4 binary sequences, i.e., $\{000, 001, 010, 100\}$, with the probability distribution $\{6/9, 1/9, 1/9, 1/9\}$. Note that the possible outcomes are dependently distributed, yet not identical. The corresponding joint entropy of X_r is 1.45.

Joint entropy of independent links

For the links associated with different inner nodes, they are independent random variables. The joint ensemble of independent links is the sum of the link entropy of each link, i.e.,

$$H(X) = H(A_{i_1 j_1}^{r_1} A_{i_2 j_2}^{r_2} \dots A_{i_H j_H}^{r_H}) = \sum_{b=1}^H H(A_{i_b j_b}^{r_b}) \quad (3.10)$$

Joint entropy of arbitrary links

Given a set of arbitrary links, we separate them into two categories: independent links and dependent links. Essentially we arrange the links in different groups according to the inner node in the dendrogram they associate to. Links in the same group are dependent. Otherwise, they are independent. The joint entropy of arbitrary links is the sum of the joint entropy of each group, which is given by the following equation,

$$H(X_{r_1, r_2, \dots, r_U}) = \sum_{u=1}^U H(X_{r_u}) \quad (3.11)$$

In order to provide a more meaningful measure of each vertex's privacy, we next define the notation of the entropy of a vertex's egocentric network. The egocentric network is one smallest subgraph centered on each node. The egocentric network entropy is an ensemble entropy regarding all the immediate links associated with the node,

Definition 3.1. (*Joint entropy of vertex i 's egocentric network*) $H(v_i)$ is the entropy of the joint link entropy $H(A_{i_1} A_{i_2} \dots A_{i_{n-1}})$ which includes all links incident to vertex i .

This definition quantifies an attacker's uncertainty towards the composition of one vertex's egocentric network.

Traditionally, for randomization schemes, the posterior belief is used to measure an attacker's uncertainty [YW08; HMJ+08]. Here we use entropy rather than the posterior belief for clarifying the ensemble uncertainty of possible worlds. Consider a link with probability p . For an attacker, there are two scenarios: $A_{ij} = 0$ or $A_{ij} = 1$. Rather than specifying that the attacker has posterior belief p for $A_{ij} = 1$ and posterior belief $1 - p$ for $A_{ij} = 0$, we use $H(A_{ij})$ to evaluate the attacker's uncertainty of this link random variable as a whole. $H(A_{ij})$ reflects the extent to which an attacker is unsure of A_{ij} 's real outcome in all its possible worlds. Note that the possible worlds are not always evenly distributed. Entropy describes the extent of obfuscation compactly instead of specifying several probabilities of each possible world. This is particularly convenient in more intricate scenarios, especially in the case of joint entropy.

Essentially, the released graph conveys the same amount of information contained in the dendrogram, which indicates that the most amount of information attackers

can infer from one released graph is just the dendrogram, by using MCMC to learn from the released graph. Note that each link replacement procedure associated with one inner node in the dendrogram cannot be directly learned, since it operates as a non-deterministic mapping function. Information transferred after link obfuscation is inherently blurred according to the HRG model.

3.5.2 Link Obfuscation VS Node Obfuscation

In [BGT11], Bonchi et. al. claim that entropy-based quantification of anonymity level is more adequate than quantification based on posteriori belief probabilities. Our approach is similar in spirit to their work, but differs crucially in the quantity under measurement. Rather than defining the quantity of node identity anonymity level directly, we consider an entropy quantification for links. Bonchi’s work is mainly concerned with re-identification of node identity, while our work attempts to address re-identification of links. Moreover, in [BGT11], node candidates are the vertices in the released graph \tilde{G} . But in our scheme, link candidates are the imaginary possible worlds during obfuscation scheme.

Furthermore, we show that link entropy distinguishes the uncertainty of links in different distributions of possible worlds under randomization scheme. As illustration, we consider the following two situations:

$$\begin{array}{ll}
 (1) \ p(A_{ab} = 0, A_{ac} = 1) = 1/2, & (2) \ p(A_{ab} = 0, A_{ac} = 1) = 1/2, \\
 \quad p(A_{ab} = 1, A_{ac} = 0) = 1/2, & \quad p(A_{ab} = 1, A_{ac} = 0) = 1/6, \\
 \quad p(A_{ab} = 0, A_{ac} = 0) = 0, & \quad p(A_{ab} = 0, A_{ac} = 0) = 1/6, \\
 \quad p(A_{ab} = 1, A_{ac} = 1) = 0. & \quad p(A_{ab} = 1, A_{ac} = 1) = 1/6.
 \end{array}$$

We note that with the probabilities listed above, it is hard to evaluate which of the two cases brings more uncertainty. We next show how link entropy can distinguish the extent of these two cases’ uncertainty. According to Equation 7, in case 1, the joint link entropy is $\log_2 2$, but it is $\log_2 2\sqrt{3}$ in case 2. Although an attacker’s greatest confidence about the state of links in released graph is both $1/2$ in two the cases, the attacker needs more effort to cross out the more uncertain possible worlds in the second case. Under LORA, during link replacement, the existence of a weak tie in graph is blurred since the link probability is effectively being spread among the fake candidate links in the

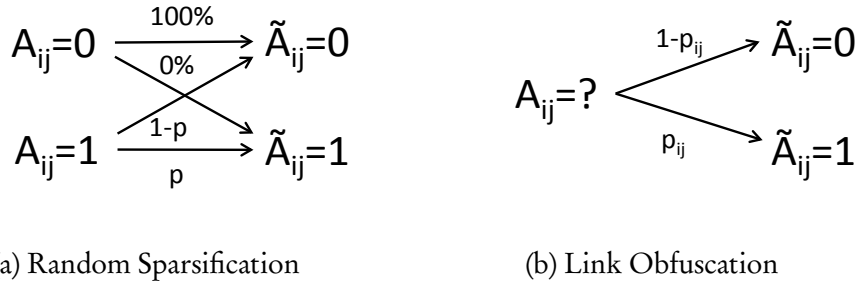


Figure 3-3: Link Obfuscation VS Random Sparsification

equivalence class. Thus, the uncertainty of possible worlds of links is increased.

Our scheme is specially designed for link privacy in the first place. But more importantly, the uncertainty of links would directly undermine the structural knowledge that the attackers can hold in any attacks. This is because links, the smallest atomic elements in graph, are the foundations of all the structure knowledge attackers can hold in a simple graph.

3.5.3 Randomization by Link Obfuscation VS Edge Addition/Deletion

Unlike randomization schemes in [HMJ+08; YW08; HGP09; BGT11], link probability obfuscation is a sophisticated method based on the source graph’s characteristics. We use Figure 3-3 to illustrate the difference between random sparsification [BGT11] and link obfuscation. For the pure random sparsification, links are perturbed in a way similar to a coin flipping game, where the coins are the same and independent. As it turns out in Figure 3-3a, every A_{ij} is associated with the same parameter p in the procedure of perturbation. Each A_{ij} “flips” like a coin in the same way. Conversely, in LORA (see in Figure 3-3b), each A_{ij} owns its specific perturbation parameter p_{ij} . Each inner node in the dendrogram is associated with one independent link replacement procedure. During each procedure, links in the link equivalence class are all related. This means more dedicate modifications are allowed on the source graph.

From entropy’s perspective, in the former scheme, all $A_{ij} = 0$ will retain its state in the released graph. Therefore, the entropy $H(A_{ij}|\tilde{A}_{ij} = 1) = -p(A_{ij} = 0|\tilde{A}_{ij} = 1) \log p(A_{ij} = 0|\tilde{A}_{ij} = 1) - p(A_{ij} = 1|\tilde{A}_{ij} = 1) \log p(A_{ij} = 1|\tilde{A}_{ij} = 1) = -0 \cdot \log 0 - 1 \cdot \log 1 = 0$. This implies that an attacker can learn that $A_{ij} = 1$ if the observation is $\tilde{A}_{ij} = 1$ in the released graph. However, in the latter scheme, p_{ij} that A_{ij} associates

with is not necessarily always 0 or 1. Hence, if $p_{ij} \neq 0$ or $p_{ij} \neq 1$, it is almost not learnable from the obfuscation procedure. In other words, it is difficult to infer the true state in the source graph with full confidence.

3.6 Experimental Studies

In this section, we report empirical results of experimental studies to evaluate the effectiveness of LORA.

3.6.1 Datasets

We conducted our experiments on four real-world datasets, namely *polblogs*, *wiki-Vote*, *ca-HepPh* and *ca-AstroPh*¹.

polblogs A network recorded in 2005 that contains of hyperlinks between weblogs on US politics.

wiki-Vote A social network contains Wikipedia voting information for adminship elections. An edge is created between two participants if one voted on or was voted by the other.

ca-HepPh A collaboration network depicts scientific collaborations between authors whose papers submitted to High Energy Physics category. An edge is created if two authors co-authored a paper.

ca-AstroPh A collaboration network covers collaboration between authors whose papers submitted to Astro Physics category.

A detailed record about the number of vertices and the number of edges in the above networks is provided in Table 3.1. All datasets are pre-processed to be undirected without self-loops.

3.6.2 Experimental Setup

One objective of our experiments is to evaluate to which extent our method can preserve the network data’s features. To this end, we measure several network statistics

¹*polblogs* is available at <http://www-personal.umich.edu/~mejn/netdata/>; *wiki-Vote*, *ca-HepPh* and *ca-AstroPh* are available at <http://snap.stanford.edu/data/index.html>.

Table 3.1: Network dataset statistics

Dataset	#Nodes	#Edges	Max Degree Pair
<i>polblogs</i>	1,224	16,715	(351, 277)
<i>wiki-Vote</i>	7,115	100,762	(1065, 773)
<i>ca-HepPh</i>	12,008	118,489	(491, 486)
<i>ca-AstroPh</i>	18,772	198,050	(504, 420)

both on the original networks and on the anonymized networks. We then compare the results and expect to observe that the anonymized networks are similar to the original ones regarding to such statistics. Specifically, we will examine the networks’ degree distribution (i.e., the distribution of vertices degrees over the whole network), shortest path lengths (i.e., the shortest path lengths between any two vertices in the network) and compare the composition of top-k influential vertices (including overlap percentage and mean absolute error of the vertex’s influence score).

Our second objective is to assess the level of anonymity by our method. Our strategy is to measure the egocentric entropy for all vertices in the source graph of each dataset, according to the derived best-fitting HRG model used to reconstruct the released graph. Notice that the entropy we measure here is a subgraph’s entropy, that is, each node’s egocentric network. The larger the entropy is, the more uncertainty possesses by the HRG structure. The lowest value of entropy is zero. It means there is no uncertainty regarding the node’s egocentric network.

All our experiments were done on Intel Xeon E5607 servers with 2.27G CPU and 32GB RAM.

3.6.3 Data Utility Analysis

Now we report the experiments on network statistics for the four datasets.

Degree Distribution

Figure 3-4 shows the degree distribution histogram of the source and released graphs. We have on the x -axis the degree size and on the y -axis in log-scale the count of vertices having the corresponding degree. We observe that in each subfigure, the two histograms share similar shapes. Specifically, we see the released graph can still mimic the original graph on the large degrees after the perturbation. This indicates both histograms have “fat tails” with similar length and shapes in their distributions. It is

known that such networks characterized by the fat-tail have scale-free property.

Shortest Path Lengths

Figure 3-5 reports shortest path length histograms. As it turns out, the released graphs' distribution does not lose much of the general shape of source graphs in all figures. In most figures, we also observe an increase in small short path in the released graphs. However, it should be noted that often short paths correspond to the local structures in a network. Hence we believe this shall not significantly affect the network's global structure.

Influential Vertices

Identifying the most influential vertices/nodes in social networks is a key problem in social network analysis. In our experiments, we consider the *eigenvector centrality* (EVC) score as the measure to rank the vertices in networks. EVC scores correspond to the values of the first leading eigenvector of the graph's adjacency matrix (the one with the greatest eigenvalue). Intuitively, EVC measures the nodes' influence by virtue of their positions in a network, that is, the sum of the geodesic distances from each vertex to all others. The eigenvector approach attempts to find the most central vertices (i.e., those with the smallest geodesic distance to others) in terms of the "global" or "overall" structure of the network. The first eigenvector usually captures the "global" aspects of distances among vertices, while the second and subsequent ones capture more specific and local structures.

In our experiments, we first test the percentage of common vertices in top- k most influential vertices of the original graphs and those of the sanitized graphs. We examine top 10, 20, 50 influential vertices as well as top 1% and 5% nodes in the networks. The results are presented in Figure 3-6. We see that *HRG* guarantees a consistent 25%-75% overlap of common vertices in most cases.

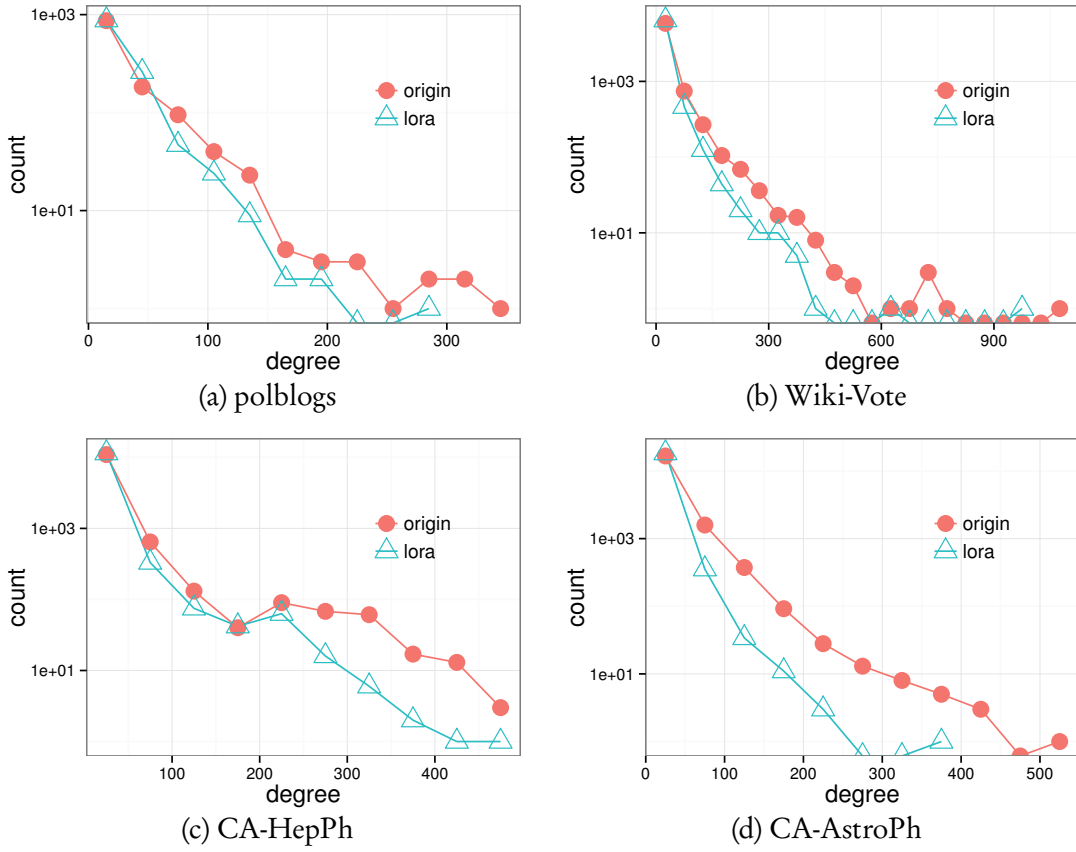


Figure 3-4: Degree distribution

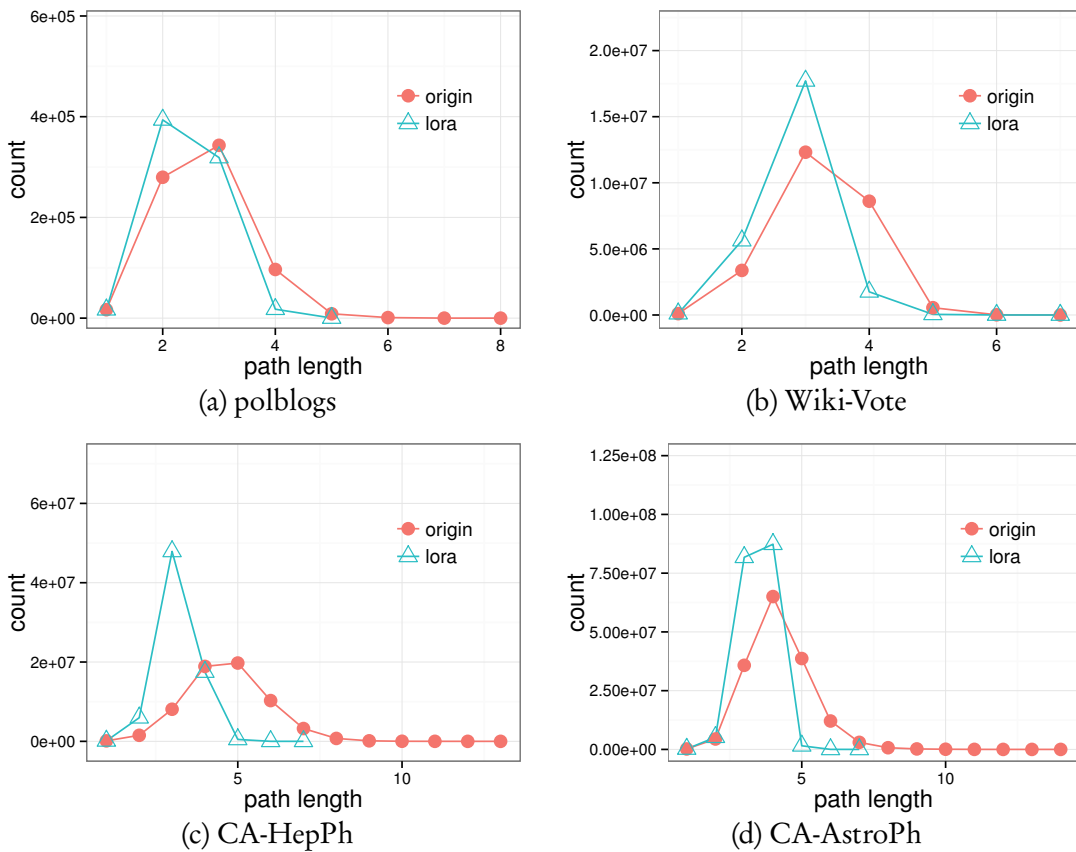


Figure 3-5: Shortest Path Distribution

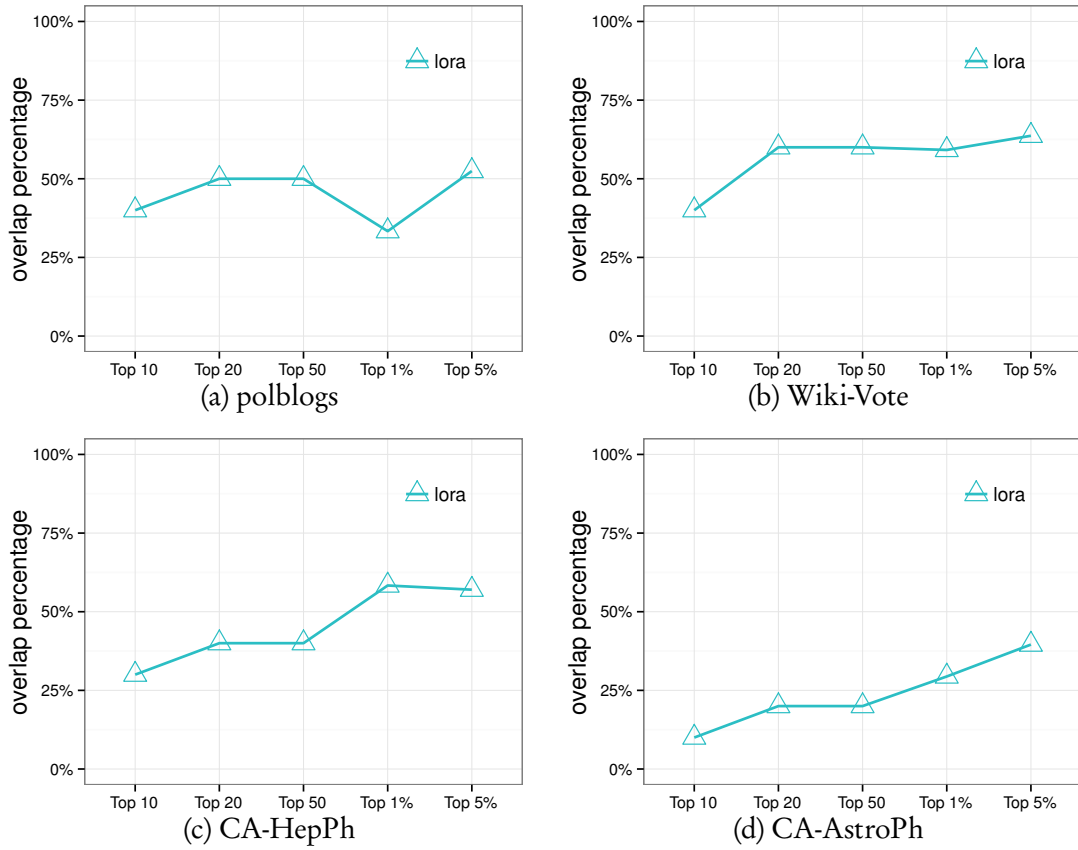


Figure 3-6: Overlap percentage of top-k influential vertices

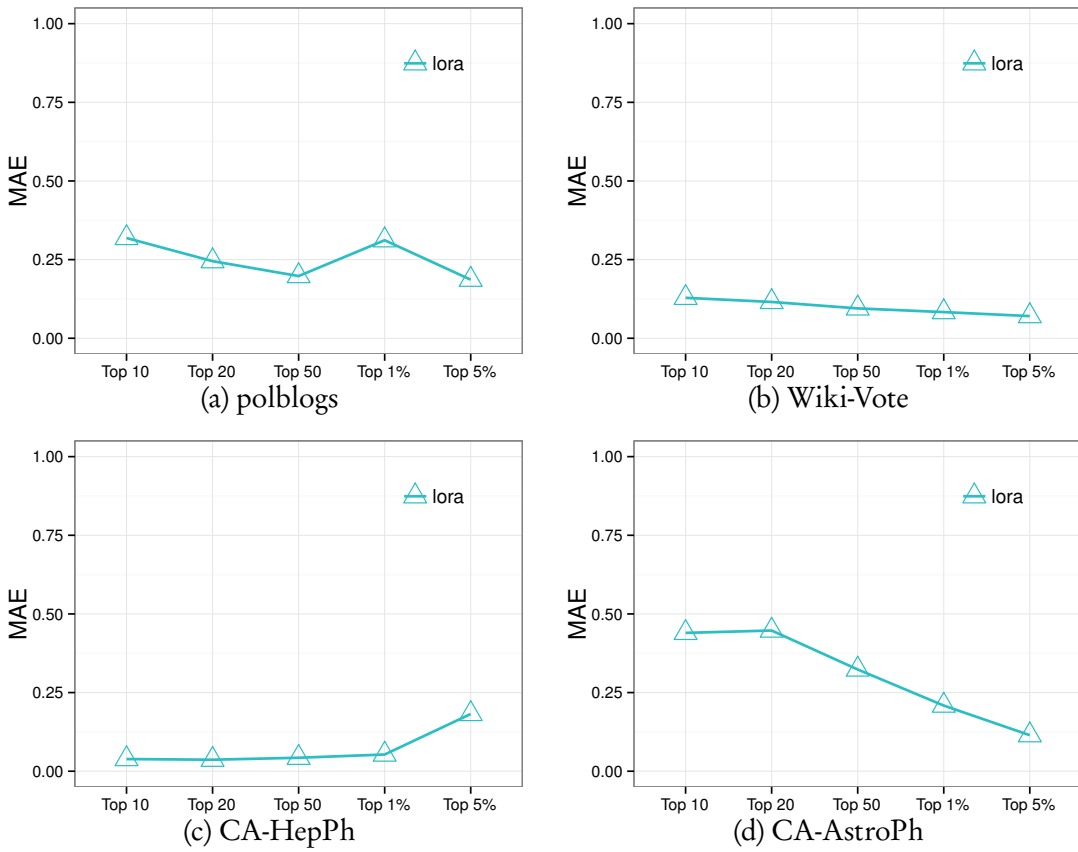


Figure 3-7: Mean absolute error of top-k vertices

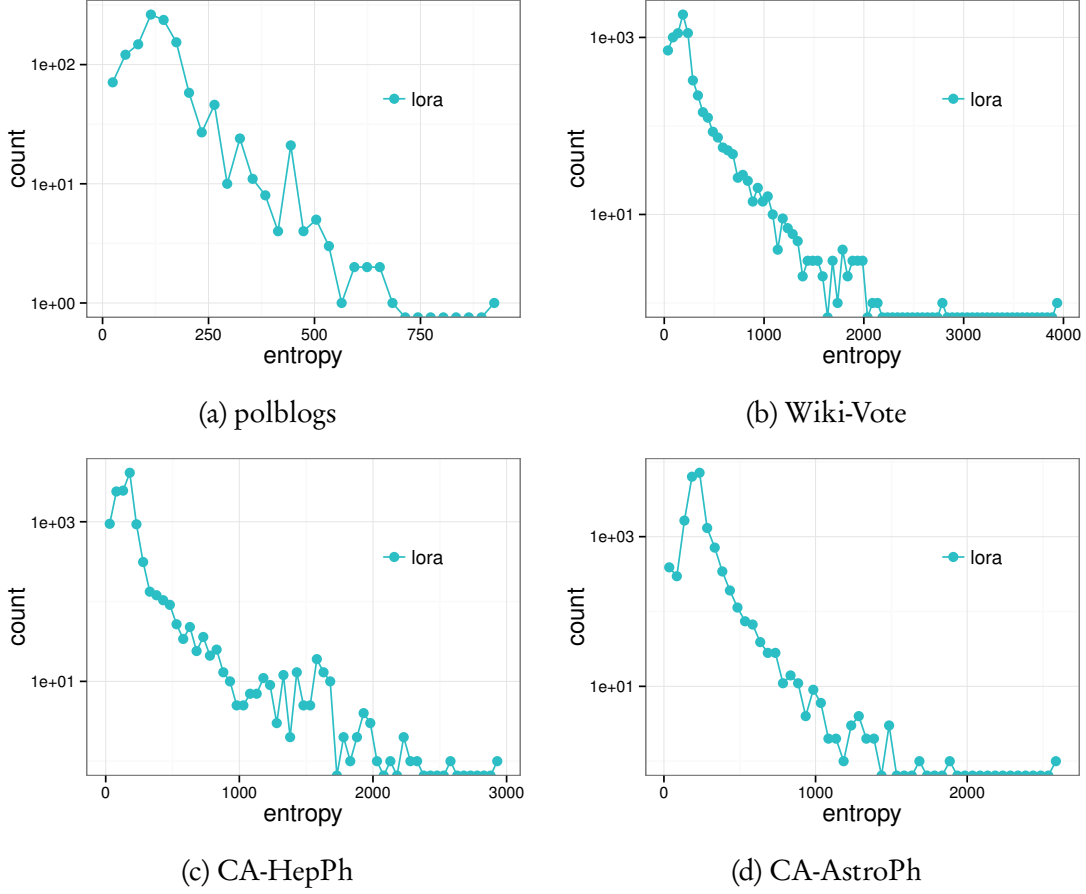


Figure 3-8: Ego-centric entropy

We then calculate the mean absolute error of the top- k most influential vertices' EVC scores. Let the set of top- k vertices in the original graph be α and that of the sanitized graph be β . To show that nodes in β have similar centralities to those in α , we use the *mean absolute error* (MAE) to compare the EVC scores in β with those in α . Formally, the MAE value is formulated by:

$$MAE(\alpha, \beta) = \frac{1}{k} \sum_{i=1}^k \left| EVC(v_{\alpha}^i) - EVC(v_{\beta}^i) \right|, \quad (3.12)$$

where v_{α}^i and v_{β}^i are the top- i nodes in α and β , respectively. The result is shown in Figure 3-7. We see that all MAE value remains low (around or below 0.25) for most cases.

3.6.4 Privacy Analysis

To understand the effectiveness of LORA, we report the histogram of the vertex egocentric entropy in Figure 3-8. The horizontal axis specifies the egocentric entropy interval. The vertical axis specifies the number of vertices. From Figure 3-8, we observe that a great portion of vertices has very large entropy (in unit of bit). It indicates significant amount of uncertainty regarding these vertices' egocentric networks in the released graph. However, we also note that there are some vertices with relatively low entropy. In this case, attackers can believe their egocentric networks in the released graph are likely to be close to the source egocentric networks. However, it should be noted that such nodes with low egocentric entropy can be further generalized in order to improve their privacy. As link probabilities of such nodes are close to 0 or 1, they are most likely to be the nodes in cliques or isolated. Clique-like nodes can be coalesced into a super node. Figure 3-2 shows one such example. Isolated nodes can be removed from the released graphs. Both procedures shall not significantly disrupt the global graph structure. Such node generalization strategy is akin to the approach illustrated in [HMJ+08].

3.7 Summary

In this chapter, we have proposed a randomization scheme, LORA, to preserve link privacy of network data publishing. LORA builds the best-fitting HRG structure of the source graph, and uses it to reconstruct a set of graphs that preserve the statistical graph properties of the source graph. The released graph is then selected from these graphs. We also introduced and argued that the *link entropy* concept is an appropriate measure of the uncertainty degree of links. Our experimental results showed that the released (reconstructed) graphs have acceptable link entropy while preserving statistical properties such as degree distribution, shortest path lengths and influential nodes.

Chapter 4

Differentially Private Network Data Release via Structural Inference

4.1 Introduction

Previously, a great deal of work has investigated *anonymization* techniques [ZP08; LT08; CSY+08; HMJ+08; ZCO09; CFL10] to ensure network data privacy. Our first work on LORA in Chapter 3 also can be categorized into this group. However, it has been shown that anonymization is susceptible to several newly discovered privacy attacks and might lead to further privacy breaches. Recently, *differential privacy* (DP) [DMN+06] has been proposed to solve such vulnerability. In this chapter, we study the problem of releasing network data under this emerging privacy standard. Given a network dataset, our goal is to release its sanitized differentially private version to hide each participant’s connections to others while preserving essential structural information to support data analysis. In this work, we adopt the rigorous differential privacy definition, that is, ϵ -differential privacy, to be our privacy model.

To ensure differential privacy, the standard technique is to add Laplace noise to query answers. However, network data can be very sensitive to relatively small changes in the network structure. Direct perturbation in the data domain (e.g., adding noise to a subgraph counting query in order to obscure the presence or absence of an edge) normally incurs excessive noise, which makes it impossible to conduct any effective data mining on the sanitized data. An alternative solution is to first project the data to other domains (e.g., the graph spectral domain [WWW13], which is analogous to

the classical frequency domain, or some parametric model space that describes the observed network, such as dK -2 series [SZW+11; WW13]). While this idea is appealing, the resultant data utility of the existing works in this direction is still undesirable for many graph mining algorithms. For example, Wang et al. [WWW13] propose to perturb the eigenvalues and eigenvectors of the corresponding adjacency matrix. This approach requires to impose noise of magnitude proportional to $O(\sqrt{n})$, where n is the number of vertices in the input network, and therefore massive noise has to be injected in large real-life network datasets. As another example, the works [SZW+11; WW13] consider to approximate the original network by the dK -series. To achieve ϵ -differential privacy, the global sensitivity of this scheme is $O(n)$ even for dK -2 series, which also demands excessive noise to be added.

In this work, we advocate a different approach that can offer better data utility. Broadly, we propose to encode a network’s structural information in terms of *link probabilities* between vertices, rather than the presence or absence of the observed edges. The fundamental advantage of adopting such a perspective is that we can capture the generally understandable and statistically meaningful properties of the network while “diluting” the impact of a single edge. In the context of differential privacy, this means that we can significantly lower the magnitude of noise added to mask the change of a single edge.

In essence, link probabilities can be estimated by a set of edge-counting queries (i.e., a query that counts the number of edges between two given sets of vertices). Therefore, our problem can be converted to a problem of finding a strategy to identify a good set of edge-counting queries that truthfully represent a network’s structure. This can be done in many possible ways. In particular, we use the statistical *hierarchical random graph* (HRG) model [CMN08], the same graph model adopted in Chapter 3, for this purpose. Recall that HRG model can carefully map all participants of a network into a hierarchical structure (called a *dendrogram*) and record link probabilities between any pair of vertices in the network. This allows us to draw a sample model from the model’s space, which essentially consists of a set of good edge-counting queries. Moreover, the model itself is paired with a likelihood score, which makes it possible to observe the quality of released data.

Technically, we make the following contributions. Unlike existing studies, we

propose to infer a network’s structure via link probabilities. We further identify that the HRG model can be used to encode a network in terms of a set of such link probabilities. Generating a good HRG under differential privacy requires careful design. We do not directly perturb the best-fitting HRG of the input network (i.e., the HRG generated by the non-private algorithm), but rather, we infer the HRG by learning in the entire HRG model space and sampling an HRG by a Markov chain Monte Carlo (MCMC) method while satisfying differential privacy. Given a sampled HRG, we propose a carefully designed thresholding strategy coupled with the Erdős-Rényi model to calculate the noisy link probabilities.

We adopt such a methodology for two reasons. First, relying on the best-fitting HRG itself will incur a high sensitivity. Changing even one edge in the network may result in a great number of changes in both the dendrogram’s structure and the set of its associated link probabilities. This is undesirable since it may alter many of the HRG’s parameters in the worst case. In contrast, we design an MCMC method to iteratively learn a reasonably good HRG from the entire HRG space. By construction, with a single edge difference, only one probability in the HRG would be influenced. Second, it is non-trivial to sample a good HRG in our setting because it is computationally challenging to compute the scores of all possible HRGs even for a small network. It can be seen that there are a total of $(2n - 3)!! \approx \sqrt{2}(2n)^{n-1}e^{-n}$ possible dendrograms for a network with n vertices. Hence, it is computationally infeasible to directly apply the exponential mechanism. We side-step this problem by using an MCMC method, which is in a similar spirit to the idea in [SY13]. However, our problem and challenges are quite different from those in [SY13]. Our goal is to publish the entire graph, *not* frequent subgraphs. A direct consequence is that we have to harness the large sensitivity in our problem, while it is always 1 in [SY13].

From the perspective of utility, we rigorously prove that the sensitivity of our proposed approach is $O(\log n)$ for fitting the dendrogram structure, which reaps the benefit of preserving good data utility in theory. We conduct extensive experiments on four real-life datasets to evaluate the effectiveness of our solution. We demonstrate that our approach significantly outperforms the state-of-the-art competitors [WW13; WWW13].

4.2 Preliminaries

In this section, we briefly introduce the hierarchical random graph (HRG) model and differential privacy.

4.2.1 Hierarchical Random Graph

HRG was previously introduced in Chapter 3 where we presented the work on LORA. But in LORA, we put our focus on the best-fitting HRG structure. In this section, we will elaborate more about the distribution of HRG in the entire structure space.

Basically, we view the underlying network in observation as a data sample drawn from a population of networks of interest. HRG essentially describes a population of random graphs that share similar topological structure. To obtain the best-fitting HRG that matches the observed graph in its structure features, we start with a random HRG, sample HRG in sequence by performing a random walk over the entire HRG space, continually making inferences from data and gradually reducing the uncertainty. On the other hand, Clauset et al. [CMN07; CMN08] show that such hierarchical structure can be an inherent core aspect of networks. Hence HRG can reproduce simultaneously many important statistical features and signature network behaviors in its random graph samples after the convergence. As a theoretical tool yet still affording great flexibility, HRG has recently begun to gain recognition and been included in the popular network analysis library, “igraph”¹.

For ease of reference, we reproduce the notations in Chapter 3 here. Let $G = (V, E)$ to be the original network data (i.e., an undirected simple graph) we want to release. T denotes a HRG’s structure, that is, a *dendrogram*. T is essentially a rooted binary tree with n leaf nodes corresponding to the n vertices of G . Each internal node r of T is associated with a probability p_r . For any two vertices (i, j) of G , their probability of being connected $p_{i,j} = p_r$, where r is their lowest common ancestor in T . Formally, an HRG is defined by a pair $(T, \{p_r\})$.

L_r and R_r denote the left and right subtrees of r respectively. n_{L_r} and n_{R_r} denote the numbers of leaves in L_r and R_r respectively. We use e_r to represent the number of edges in G whose endpoints are leaves of each of the two subtrees of r in T .

¹<http://igraph.org/c/doc/igraph-HRG.html>

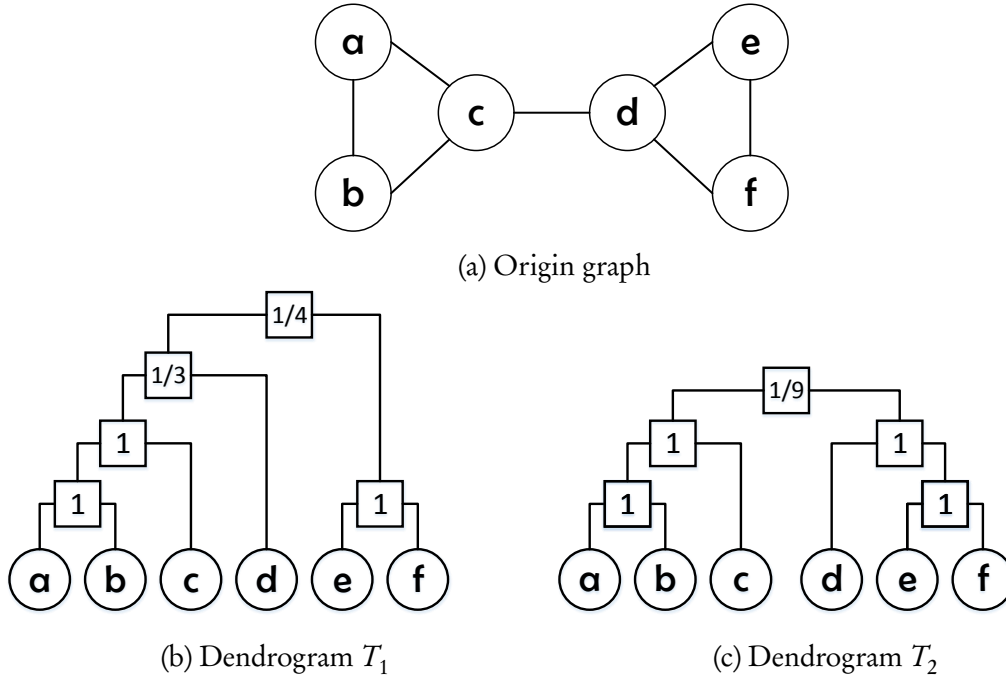


Figure 4-1: An example of the HRG model in [CMN08]

The *log-likelihood* of an HRG for a given graph G measures how plausible this HRG is to represent G , which can be computed with the following formula:

$$\log \mathcal{L}(T, \{p_r\}) = - \sum_{r \in T} n_{L_r} n_{R_r} b(p_r) \quad (4.1)$$

where $b(p_r) = -p_r \log p_r - (1-p_r) \log(1-p_r)$ is the Gibbs-Shannon entropy function. We use $\mathcal{L}(T)$ to simplify the notation $\mathcal{L}(T, \{p_r\})$ in the following context when there is no confusion.

We now use one example to further demonstrate how the *log-likelihoods* distinguish different HRG structures in terms of their fitness to describe the underlying networks.

Example 4.1. Figure 4-1b and 4-1c give an example of two possible dendrograms, T_1 and T_2 , for an original graph in Figure 4-1a (the same graph used in Chapter 3). Following the same method in Chapter 3, it's easy to calculate all $\{p_r\}$. We then compute the likelihoods of the dendrogram T_1 and T_2 . Specifically, $\mathcal{L}(T_1) = (1/3)(2/3)^2(1/4)^2(3/4)^6 \approx 0.00165$, and $\mathcal{L}(T_2) = (1/9)(8/9)^8 \approx 0.0433$. Since $\mathcal{L}(T_2)$ is much larger than $\mathcal{L}(T_1)$, T_2 is a more plausible hierarchy to describe the original graph. As we can see visually, this is indeed the case. ■

4.2.2 Differential Privacy

Differential privacy [DMN+06] has emerged as a prevalent privacy model. It is based on the concept of *neighboring* databases. The privacy guarantee of differential privacy in the context of network data depends on the interpretation of *neighboring* graphs. In this work, we define two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ to be *neighbors* if $V_1 = V_2$, $E_1 \subset E_2$ and $|E_1| + 1 = |E_2|$. Formally, ϵ -differential privacy for network data is defined below.

Definition 4.1 (ϵ -Differential privacy). *A randomized algorithm \mathcal{A} is ϵ -differentially private if for any two neighboring graphs G_1 and G_2 , and for all outputs $O \subseteq \text{Range}(\mathcal{A})$,*

$$\Pr[\mathcal{A}(G_1) \in O] \leq e^\epsilon \times \Pr[\mathcal{A}(G_2) \in O] \blacksquare$$

Our definition of differential privacy is also known as *edge differential privacy* [HLM+09]. Intuitively, it hides the existence of *any* single edge from an adversary. The smaller ϵ is, the better the privacy protection is. Normally, ϵ is a small value (e.g., $\epsilon \leq 1$).

Differential privacy can be achieved by two standard mechanisms, the *Laplace mechanism* [DMN+06] and the *exponential mechanism* [MT07]. Both mechanisms are based on the concept of *global sensitivity* of a function f . For any two neighboring graphs G_1 and G_2 , the global sensitivity of a function $f : G \rightarrow \mathbb{R}^d$ is defined as $\Delta f = \max_{G_1, G_2} \|f(G_1) - f(G_2)\|_1$, where d is the metric on the output space (we use L1 distance as the metric in our definition).

The Laplace mechanism is mainly used for queries which return real values. It adds properly calibrated noise to the true answer to a query. More precisely, given a function f and the privacy parameter ϵ , the noise is drawn from a Laplace distribution with the probability density function $p(x|\lambda) = \frac{1}{2\lambda} e^{-|x|/\lambda}$, where $\lambda = \Delta f / \epsilon$.

Theorem 4.1 (Laplace mechanism [DMN+06]). *For any function $f : G \rightarrow \mathbb{R}^d$, the mechanism \mathcal{A}*

$$\mathcal{A}(G) = f(G) + \langle \text{Lap}_1\left(\frac{\Delta f}{\epsilon}\right), \dots, \text{Lap}_d\left(\frac{\Delta f}{\epsilon}\right) \rangle$$

gives ϵ -differential privacy, where $\text{Lap}_i\left(\frac{\Delta f}{\epsilon}\right)$ are i.i.d Laplace variables with scale parameter $\frac{\Delta f}{\epsilon}$. \blacksquare

The exponential mechanism is mainly used for functions whose outputs are not real numbers. Its general idea is to sample an output o from the output space \mathcal{O} according to a utility function u . It assigns exponentially greater probabilities of being selected to outputs of higher scores so that the final output would be close to the optimum with respect to u . Let the global sensitivity of u be $\Delta u = \max_{o, G_1, G_2} |u(G_1, o) - u(G_2, o)|$.

Theorem 4.2 (Exponential mechanism [MT07]). *Given a utility function $u : (G \times \mathcal{O}) \rightarrow \mathbb{R}$ for a graph G , the mechanism \mathcal{A} that samples an output o with probability proportional to $\exp(\frac{\epsilon \cdot u(G, o)}{2\Delta u})$ satisfies ϵ -differential privacy. ■*

4.3 Structural Inference under Differential Privacy

4.3.1 Overview

Before presenting the details, we first give an overview of our method. Our goal is to release a sanitized network \tilde{G} that matches the structural properties of the original network G as closely as possible while satisfying ϵ -differential privacy. Our general idea is to identify the hierarchical random graph (HRG) that best fits G and then generate \tilde{G} from the identified HRG.

Recall that an HRG consists of a dendrogram T and a set of associated probabilities $\{p_r\}$. This means that we need to not only identify a good fitting dendrogram but also calculate its associated probabilities. In this process, we face two major technical challenges: (1) How to find a good dendrogram from a factorial number of candidates while satisfying ϵ -differential privacy, and (2) how to calculate the probabilities that might be dominated by injected noise. We address the first challenge by designing a Markov chain Monte Carlo (MCMC) procedure, which samples a good dendrogram according to its likelihood. Next, we cope with the second challenge by developing an effective thresholding strategy that is backed up by the Erdős-Rényi model. After generating a representative HRG for G , we generate \tilde{G} by placing edges according to $\{p_r\}$.

Algorithm 4.1: Differentially Private Dendrogram Fitting

Input : Input graph G , privacy parameter ϵ_1

Output: Sampled dendrogram T_{sample}

- 1 Initialize the Markov chain by choosing a random starting dendrogram T_0 ;
 - 2 **for** each step i of the Markov chain **do**
 - 3 Randomly pick an internal node r in T_{i-1} ;
 - 4 Pick a neighboring dendrogram T' of T_{i-1} by randomly drawing a configuration of r 's subtrees;
 - 5 Accept the transition and set $T_i = T'$ with probability $\min\left(1, \frac{\exp(\frac{\epsilon_1}{2\Delta u} \cdot \log \mathcal{L}(T'))}{\exp(\frac{\epsilon_1}{2\Delta u} \cdot \log \mathcal{L}(T_{i-1}))}\right)$;
 - 6 **end**
 - 7 //when equilibrium is reached
 - 8 **return** the sampled dendrogram $T_{sample} = T_i$;
-

4.3.2 Algorithms

We now formally describe our solution (referred to as *HRG* in the sequel). Our solution is composed of three steps: (1) differentially privately sample a good dendrogram T_{sample} from the entire dendrogram space (Algorithm 4.1); (2) given the sampled dendrogram T_{sample} , compute the probabilities $\{p_r\}$ associated with T_{sample} (Algorithm 4.2); (3) generate the sanitized graph according to the identified HRG (Algorithm 4.3). We divide the total privacy parameter ϵ into 2 portions, ϵ_1 and ϵ_2 , each being used in one of the first two steps. Note that the third step does not require any privacy parameter.

Differentially Private Dendrogram Fitting. Since, for an input graph G with n vertices, each of its dendrograms T is associated with a log-likelihood $\log \mathcal{L}(T)$, which measures its goodness of representing G , a straightforward attempt to achieve differential privacy is to employ the exponential mechanism. Let the utility function be $u(T) = \log \mathcal{L}(T)$. The exponential mechanism samples T with probability proportional to $\frac{\exp(\frac{\epsilon_1}{2\Delta u} \cdot u(T))}{\sum_{T' \in \mathcal{T}} \exp(\frac{\epsilon_1}{2\Delta u} \cdot u(T'))}$, where \mathcal{T} is the entire output space (i.e., the set of all possible dendrograms of G). Unfortunately, this simple idea is computationally infeasible because it requires to enumerate a total of $|\mathcal{T}| = (2n - 3)!! \approx \sqrt{2}(2n)^{n-1}e^{-n}$ possible dendrograms. In our solution, we overcome the issue by designing an MCMC process, which simulates the exponential mechanism by a sequence of *local* transitions

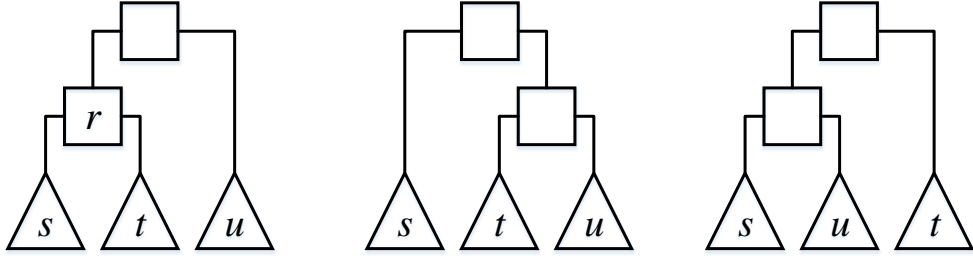


Figure 4-2: Three configurations of r 's subtrees [CMN08]

in \mathcal{T} . Our differentially private dendrogram fitting algorithm is summarized in Algorithm 4.1.

Algorithm 4.1 is based on the Metropolis algorithm [BGJ+11]. It starts by choosing an arbitrary dendrogram $T_0 \in \mathcal{T}$ as the initial state of the Markov chain (Line 1). It then iteratively performs the following procedure (Lines 2-6): randomly propose a neighboring dendrogram T' of the dendrogram T_{i-1} in the previous iteration and update the current state in the following way:

$$T_i = \begin{cases} T' & \text{with probability } \alpha \\ T_{i-1} & \text{with probability } 1 - \alpha \end{cases}$$

where the acceptance ratio $\alpha = \min(1, \frac{\exp(\frac{\epsilon_1}{2\Delta u} \cdot \log \mathcal{L}(T'))}{\exp(\frac{\epsilon_1}{2\Delta u} \cdot \log \mathcal{L}(T_{i-1})})$ and Δu is the global sensitivity of the utility function u . We show how to calculate Δu in Section 4.4.2.

To draw a neighbor T' of T_{i-1} uniformly at random, we first randomly choose an internal node r in T_{i-1} (other than the root) and then permute the three subtrees associated with r to generate two alternative configurations of r 's subtrees, as illustrated in Figure 4-2. One of these two configurations is chosen to be the neighboring candidate T' . Let the state space of this Markov chain be \mathcal{T} . It is easy to verify that the transitions based on this permutation scheme are both *reversible* and *ergodic* (i.e., any pair of dendrograms can be connected by a finite sequence of such transitions). Hence, such an MCMC procedure has a unique stationary distribution after it converges to equilibrium. We run the above Markov chain until equilibrium is reached, which indicates that the desired distribution has already been reached. Therefore, the sampled dendrogram T_{sample} is indeed drawn from the stationary distribution (Line 8).

In practice, there are many approaches to diagnose MCMC convergence. Here we follow the method used in [CMN07; CMN08]. Specifically, we use the heuristic of

Algorithm 4.2: CalculateNoisyProb($G, T_{sample}, r^*, \epsilon_2$)

Input : Input graph G , sampled dendrogram T_{sample} , privacy parameter ϵ_2 , internal node r^*

Output: A vector of noisy probabilities $\{\widetilde{p}_r\}$, where $r \in \{r^*, \text{all internal nodes below } r^*\}$

- 1 $\lambda_b = \frac{1}{\epsilon_2 \cdot (n_{L_{r^*}} \cdot n_{R_{r^*}})}$;
- 2 $\lambda_c = \frac{1}{\epsilon_2 \cdot ((n_{L_{r^*}} + n_{R_{r^*}})(n_{L_{r^*}} + n_{R_{r^*}} - 1)/2)}$;
- 3 **if** $\lambda_b \geq \tau_1$ and $\lambda_c \geq \tau_2$ **then**
- 4 $e_c(r^*) \leftarrow$ number of edges in the subgraph induced by all leaf nodes of the subtree rooted at r^* ;
- 5 $\widetilde{p} = \min\{1, \frac{e_c(r^*) + \text{Lap}(\frac{1}{\epsilon_2})}{(n_{L_{r^*}} + n_{R_{r^*}})(n_{L_{r^*}} + n_{R_{r^*}} - 1)/2}\}$;
- 6 **for each** r in $\{r^*, \text{all internal nodes below } r^*\}$ **do**
- 7 $\widetilde{p}_r = \widetilde{p}$;
- 8 **end**
- 9 **else**
- 10 $\widetilde{p}_{r^*} = \min\{1, \frac{e_{r^*} + \text{Lap}(\frac{1}{\epsilon_2})}{n_{L_{r^*}} \cdot n_{R_{r^*}}}\}$;
- 11 $r_L \leftarrow r^*$'s left child;
- 12 $r_R \leftarrow r^*$'s right child;
- 13 CalculateNoisyProb($G, T_{sample}, r_L, \epsilon_2$);
- 14 CalculateNoisyProb($G, T_{sample}, r_R, \epsilon_2$);
- 15 **end**

the average log-likelihood to judge whether the Markov chain has converged to the stationary distribution. We will elaborate more details of MCMC convergence time in Section 4.5.2. Additional discussion about the convergence and its mixing time can be found in [CMN07; CMN08].

Noisy Probability Calculation. In the second step, we calculate the noisy probabilities associated with T_{sample} 's internal nodes. Recall that, for an internal node r , its associated probability $p_r = \frac{e_r}{n_{L_r} \cdot n_{R_r}}$ (see Section 4.2.1). It is easy to observe that the probabilities of the internal nodes rooted in smaller subtrees (i.e., in lower levels of T_{sample}) are generally more sensitive to Laplace noise injected. Indeed, according to our experiments, the direct application of the Laplace mechanism to these nodes' probabilities results in poor utility. To relieve such negative effects, we propose a carefully designed thresholding strategy coupled with the Erdős-Rényi model, which is presented in Algorithm 4.2.

The general idea is that if a probability p_r cannot be “reliably” estimated by applying the Laplace mechanism to $\frac{e_r}{n_{Lr} \cdot n_{Rr}}$, we employ the Erdős-Rényi model to approximate the probability. To measure the reliability of a noisy probability, we set up the sentinel λ_b . For an internal node r^* in T_{sample} , λ_b is set to $\frac{1}{\epsilon_2 \cdot (n_{Lr^*} \cdot n_{Rr^*})}$ (Line 1), which measures the noise scale of the potential noisy probability \widetilde{p}_{r^*} . If λ_b is relatively large with respect to a threshold value τ_1 (that is, the probability cannot be reliably calculated by the Laplace mechanism), we model the subgraph induced by all leaf nodes of the subtree rooted at r^* as an Erdős-Rényi random graph. With this model, the link probability of *any* pair of vertices in this subgraph is $\frac{e_c(r^*)}{(n_{Lr^*} + n_{Rr^*})(n_{Lr^*} + n_{Rr^*} - 1)/2}$, which is later perturbed by the Laplace mechanism (Line 5). Otherwise, we can expect that $\frac{e_r}{n_{Lr} \cdot n_{Rr}}$ still gives a good estimation after adding noise. Hence we directly generate the noisy probability as $\min\{1, \frac{e_{r^*} + \text{Lap}(\frac{1}{\epsilon_2})}{n_{Lr^*} \cdot n_{Rr^*}}\}$ (Line 10) and perform the similar procedure on r^* 's children (Lines 11-14).

In Algorithm 4.2, we calculate the noisy probabilities in a top-down manner over T_{sample} . During this process, the approximated probabilities based on the Erdős-Rényi model also become less accurate due to added Laplace noise. Here, we would also like to guarantee the accuracy of the perturbed approximated probabilities. For this reason, we introduce another sentinel λ_c (Line 2), which is compared with a threshold value τ_2 to indicate whether the noise scale of the approximated probabilities is acceptable. In summary, we employ the Erdős-Rényi model when (1) the probability cannot be accurately estimated by $\frac{e_{r^*} + \text{Lap}(\frac{1}{\epsilon_2})}{n_{Lr^*} \cdot n_{Rr^*}}$ (guarded by λ_b), and (2) injecting noise to $\frac{e_c(r^*)}{(n_{Lr^*} + n_{Rr^*})(n_{Lr^*} + n_{Rr^*} - 1)/2}$ would not seriously affect its accuracy (guarded by λ_c). This explains our condition in Line 3. In this case, the probabilities of r^* and *all* internal nodes below r^* will be approximated by the Erdős-Rényi model (Lines 6-8). In our experiments, we observe that setting $\tau_1 = 0.05$ and $\tau_2 = 0.01$ gives good results over different real-life datasets. Note that the choices of these thresholds are *data-independent*: the tuning of τ_1 and τ_2 only relies on ϵ_2 .

Sanitized Graph Generation. With the sampled dendrogram T_{sample} and the set of noisy probabilities $\{\widetilde{p}_r\}$, we generate the sanitized graph as follows (Algorithm 4.3). For each pair of vertices $i, j \in V$, we find their lowest common ancestor r in T_{sample} (Line 4), and then place an edge between them in \widetilde{G} with probability \widetilde{p}_r (Line 5).

Algorithm 4.3: Generate Sanitized Graph \tilde{G}

Input : Input graph G , sampled dendrogram T_{sample} , privacy parameter ϵ_2

Output: Sanitized graph \tilde{G}

```
1  $r_{root} \leftarrow$  root node of  $T_{sample}$ ;  
2 CalculateNoisyProb( $G, T_{sample}, r_{root}, \epsilon_2$ );  
3 for each pair of vertices  $i, j \in V$  do  
4   | Find the lowest common ancestor  $r$  of  $i$  and  $j$  in  $T_{sample}$ ;  
5   | Place an edge in  $\tilde{G}$  between  $i$  and  $j$  with independent probability  $\tilde{p}_r$ ;  
6 end  
7 return sanitized graph  $\tilde{G}$ ;
```

4.4 Privacy Analysis

In this section, we formally analyze the privacy guarantee of our algorithm *HRG*.

4.4.1 Privacy via Markov Chain Monte Carlo

We first show that the MCMC-based Algorithm 4.1 can satisfy differential privacy. Recall that the main purpose of applying the MCMC method is to draw a random sample from the desired distribution. Essentially, the standard exponential mechanism for achieving differential privacy is also a method to sample an output $o \in \mathcal{O}$ in the target distribution with probability proportional to $\exp(\epsilon \cdot u(o)/2\Delta u)$, where $u(o)$ is the utility function and Δu is its sensitivity. Hence we see that, by matching the stationary distribution of MCMC with the target distribution required by the exponential mechanism, MCMC can be used to realize the exponential mechanism.

In our setting, we set the utility function $u(T)$ of a dendrogram T to be $\log \mathcal{L}(T)$, the *log-likelihood* of T , and the acceptance ratio of MCMC to be $\min(1, \frac{\exp(\frac{\epsilon_1}{2\Delta u} \cdot \log \mathcal{L}(T'))}{\exp(\frac{\epsilon_1}{2\Delta u} \cdot \log \mathcal{L}(T_{i-1})})$. Therefore, when the Markov chain converges to the stationary distribution π , we indeed draw a sample T from π with the probability mass function:

$$Pr(T) = \frac{\exp(\frac{\epsilon_1}{2\Delta u} \cdot \log \mathcal{L}(T))}{\sum_{T' \in \mathcal{T}} \exp(\frac{\epsilon_1}{2\Delta u} \cdot \log \mathcal{L}(T'))}.$$

This is equivalent to the exponential mechanism which outputs T with probability

proportional to $\exp(\frac{\epsilon_1}{2\Delta u} \cdot \log \mathcal{L}(T))$. Therefore, we can conclude that Algorithm 4.1 satisfies ϵ_1 -differential privacy.

We refer interested readers to [SY13] in which the idea of applying MCMC to achieve the exponential mechanism was first proposed for more discussion about how MCMC's stationary distribution perfectly matches the required distribution under the exponential mechanism.

4.4.2 Sensitivity Analysis

We now formally analyze the global sensitivity Δu . In this section, we will first derive how the utility function u (i.e., $\log \mathcal{L}(T)$) varies in neighboring databases. After that, we will formulate Δu and show that Δu monotonically increases as n grows. Lastly, we prove that Δu is $O(\log n)$.

In this work, we consider each possible output to be a dendrogram T in the output space \mathcal{T} . From the definition of global sensitivity, we have the following.

Definition 4.2 (Global sensitivity Δu).

$$\Delta u = \max_{T \in \mathcal{T}, G, G'} |\log \mathcal{L}(T, G) - \log \mathcal{L}(T, G')|$$

where G and G' are neighboring graphs. ■

Intuitively, Δu is the maximum change in the log-likelihood of any dendrogram in the output space if one edge is missing. It is easy to observe that missing one edge will influence exactly one internal node's probability p_r in a dendrogram. Thus, we have:

Lemma 4.1. $\Delta u = \max_{r \in \mathcal{T}} |(-n_{L_r} n_{R_r} h(p_r)) - (-n_{L_r} n_{R_r} h(p'_r))|$, where $p_r = \frac{e_r}{n_{L_r} n_{R_r}}$ and $p'_r = \frac{e_r - 1}{n_{L_r} n_{R_r}}$. ■

We now analyze how Δu varies as parameters change. Let $N = n_{L_r} \cdot n_{R_r}$. It is easy to see that there are two independent variables in Δu , the number of all possible connections N and the number of the observed edges e_r .

Theorem 4.3. Δu monotonically increases as $n \rightarrow +\infty$, and

$$\Delta u = \log N_{max} + \log\left(1 + \frac{1}{N_{max} - 1}\right)^{N_{max} - 1},$$

where $N_{\max} = \frac{n^2}{4}$ when n is even and $N_{\max} = \frac{n^2-1}{4}$ when n is odd. ■

Proof. To analyze Δu , we first fix N . Let $f(e) = h(p) - h(p')$ and $\Delta u = \max |f(e)|$. Figure 4-3(a) plots the entropy value $h(p)$ as p varies. Since $f(e)$ has the format of discrete derivative of $h(p)$, we can analyze the monotonicity of $f(e)$ by computing the second order derivative of $h(p)$. We have

$$h''(p) = -\frac{1}{1-p} - \frac{1}{p}$$

It can be observed that $h''(p) < 0$ for all p . Hence $h(p)$ is a concave function and $h'(p)$ (or the acceleration) monotonically decreases. Therefore, $f(e)$ monotonically decreases.

Since $\Delta u = \max |f(e)|$, we just need to derive the extreme values of $f(e)$. Note that $f(e) > 0$ when p is in $[0, 0.5]$ and $f(e) < 0$ when p is in $(0.5, 1]$. Hence, $\Delta u = \max(-\min(N \cdot f(e)), \max(N \cdot f(e)))$. Due to the symmetric property of $h(p)$, we can get $\max(f(e)) = -\min(f(e))$. With the monotonic property of $f(e)$, we can derive the value of Δu when $e = 1$ or $e = N_{\max}$. Next we fix $e = 1$ and vary N . Let $\Delta u = \max_{N \in [1, N_{\max}]} |f(N)|$, where

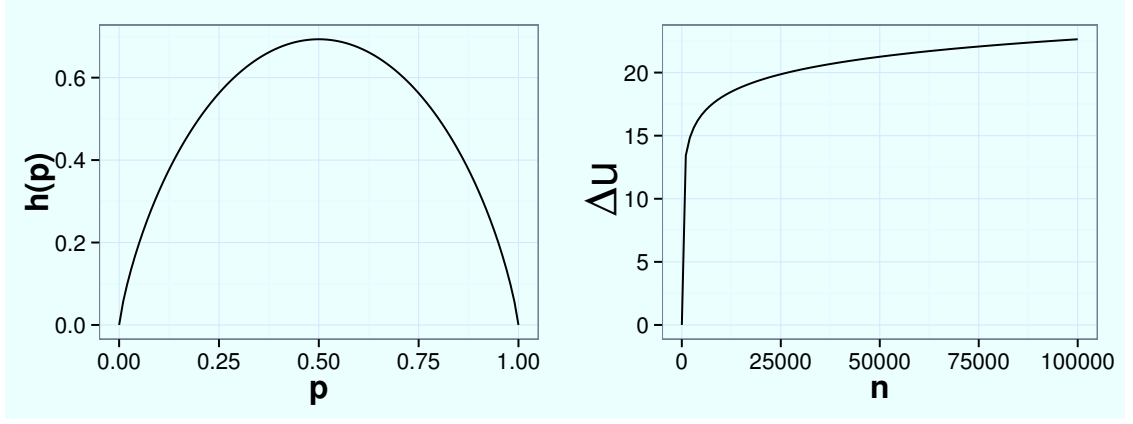
$$\begin{aligned} f(N) &= 1 \cdot \log \frac{1}{N} + (N-1) \cdot \log \left(1 - \frac{1}{N}\right) - 0 \\ &= -\log N + (N-1) \cdot \log \left(1 - \frac{1}{N}\right) \end{aligned}$$

The first order derivative of $f(N)$, $f'(N) = \log \left(1 - \frac{1}{N}\right) < 0$. Hence $f(N)$ is a decreasing function. Since $f(N) \leq 0$ for N in $[1, +\infty]$, we conclude that $\Delta u = -\min(f(N)) = -f(N_{\max})$. Hence,

$$\begin{aligned} \Delta u &= \log N_{\max} - (N_{\max} - 1) \cdot \log \frac{N_{\max} - 1}{N_{\max}} \\ &= \log N_{\max} + (N_{\max} - 1) \log \left(1 + \frac{1}{N_{\max} - 1}\right) \\ &= \log N_{\max} + \log \left(1 + \frac{1}{N_{\max} - 1}\right)^{N_{\max} - 1} \end{aligned}$$

This completes the proof. □

Next we show that Δu is $O(\log n)$, where n is the number of vertices in the input



(a)

(b)

Figure 4-3: Gibbs-Shannon entropy and plot of Δu

network.

Theorem 4.4. *The global sensitivity of a dendrogram's log-likelihood, Δu , is $O(\log n)$. ■*

Proof. Based on Theorem 4.3, we first analyze the second term of Δu , that is, $\log(1 + \frac{1}{N_{max}-1})^{N_{max}-1}$. Let $y = (1 + \frac{1}{x})^x$. We have

$$\begin{aligned} \left(1 + \frac{1}{x}\right)^x &= 1 + \binom{x}{1} \frac{1}{x} + \binom{x}{2} \frac{1}{x^2} + \binom{x}{3} \frac{1}{x^3} + \cdots + \binom{x}{x} \frac{1}{x^x} \\ &= 1 + 1 + \sum_{k=2}^n \frac{1}{k!} \frac{x(x-1)\cdots(x-(k-1))}{x^k} \end{aligned}$$

Since, for each $k \in \{2, 3, \dots, x\}$,

$$\frac{1}{k!} \frac{x(x-1)\cdots(x-(k-1))}{x^k} = \prod_{j=1}^{k-1} \left(1 - \frac{j}{x}\right)$$

which increases with x , we learn that $y = (1 + \frac{1}{x})^x$ also increases with x . As $x \rightarrow \infty$, we have $\lim_{x \rightarrow \infty} \left(1 + \frac{1}{x}\right)^x = e$. Therefore we have:

$$\begin{aligned} \Delta u &= \log N_{max} + \log\left(1 + \frac{1}{N_{max}-1}\right)^{N_{max}-1} \\ &< \log N_{max} + \log e \leq \log \frac{n^2}{4} + 1 \\ &= O(\log n) \end{aligned}$$

This completes the proof. □

Figure 4-3(b) plots the value of Δu as n increases. We see that Δu increases slowly when n becomes larger. Thus we expect that applying the exponential mechanism in terms of MCMC in this setting would guarantee good data utility even for large-scale networks.

4.4.3 Privacy via Structural Inference

Finally, we prove that our solution *HRG* is ϵ -differentially private based on the *sequential composition* property.

Theorem 4.5 (Sequential Composition [McS10]). *Let each \mathcal{A}_i provide ϵ_i -differential privacy. A sequence of $\mathcal{A}_i(D)$ over the database D provides $\sum \epsilon_i$ -differential privacy. ■*

Taken with the above theorem, we can derive that our scheme ensures ϵ -differential privacy.

Theorem 4.6. *HRG satisfies ϵ -differential privacy. ■*

Proof. We use ϵ_1 in Algorithm 4.1 for sampling the dendrogram and ϵ_2 in Algorithm 4.2 for calculating the probabilities associated with the sampled dendrogram. From the analysis in above sections, we learn that Algorithm 4.1 is ϵ_1 -differentially private. In Algorithm 4.2, we employ the Laplace mechanism to obtain the noisy answers to a set of *counting queries*. Since, by construction of a dendrogram, a single edge change will affect only one counting query by 1, Algorithm 4.2 is ϵ_2 -differentially private. Since Algorithm 4.3 is based on the differentially private HRG generated by Algorithm 4.1 and Algorithm 4.2, it does not consume any privacy budget. Hence, based on Theorem 4.6, we can conclude that our solution satisfies ϵ -differential privacy, where $\epsilon = \epsilon_1 + \epsilon_2$. □

4.5 Experimental Evaluation

In this section, we experimentally study the equilibrium of our MCMC method. We evaluate the utility of *HRG* with the same datasets used in Chapter 3, that is, *polblogs*, *wiki-Vote*, *ca-HepPh* and *ca-AstroPh*. Details about the network statistics can be found in 3.1. The experiments were done also on Intel Xeon E5607 servers with 2.27G CPU and 32GB RAM.

4.5.1 Experimental Settings

In our first set of experiments, we fix $\epsilon = 1.0$. Specifically, we assign $(\epsilon_1, \epsilon_2) = \{(0.1, 0.9), (0.5, 0.5), (0.9, 0.1)\}$ for sampling the dendrogram and computing noisy link probabilities, respectively (see Figures 4-5 – 4-8). In the second set of experiments, we study the influence of different privacy parameters on data utility. In the figures, we denote our solution *HRG* with the legend *hrg- ϵ_1 - ϵ_2* .

For comparison purposes, we implemented two state-of-the-art competitors, *spectral* [WWW13] and *dk2* [WW13]. Since no systematic approach of choosing parameter values is provided in [WWW13], we tune the parameters in *spectral* and report the best performance we obtain. More specifically, let k be the number of eigenvalues chosen in the scheme, ϵ_1 be the privacy budget for computing noisy eigenvalues and ϵ_2 for computing noisy eigenvectors. The literature [WYW+11] referred by Wang et al. in [WWW13] suggests that k is usually in the range $[2, 9]$. Hence we vary k from 2 to 9 and report the best case. In the figures, *spectral* is denoted by the legend *spec- k - ϵ_1 - ϵ_2* .

Due to the poor performance of *dk2* under ϵ -differential privacy, we compare with the scheme under a more relaxed privacy notion, that is, (ϵ, δ) -differential privacy. We follow the parameter settings in [WW13] and set $\delta = 0.01$. Unfortunately, even under (ϵ, δ) -differential privacy, we still need to use relatively large ϵ values (e.g., 200) to obtain comparable results. Moreover, the sensitivity in this case is data-dependent. It depends on the maximum degree pair in the networks (see Table 3.1). So we choose ϵ values proportional to the maximum degree pair in each network. The choice of parameters for *dk2* is denoted by the legend *dk2- ϵ - δ* .

From a privacy’s perspective, *spectral* requires the number of edges in the input network to be known, whereas our scheme *HRG* and *dk2* do not require so. In addition, *dk2* is not able to remap the nodes to the observed network, so the experiments on influential node analysis is not applicable to *dk2*.

4.5.2 Log-likelihood and MCMC Equilibrium

In practice, we diagnose MCMC’s convergence by tracing the *log-likelihood*, $\log \mathcal{L}(T)$, of the sampled dendrograms. The diagnostic takes down consecutive non-overlapping windows of the Markov chain (each window consists of 65536 MCMC steps in our experiments) and compares the means of $\log \mathcal{L}(T)$ within these windows. We use the

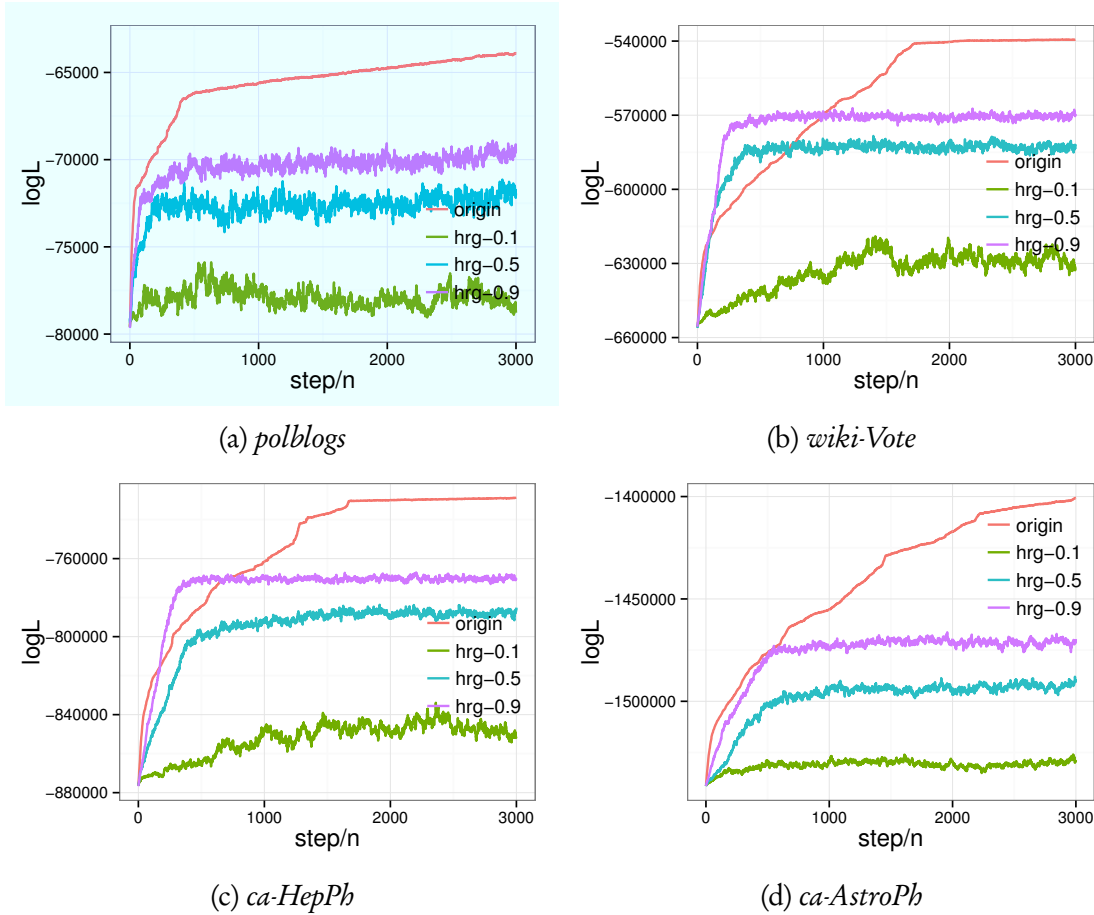


Figure 4-4: Trace of log-likelihood as a function of the number of MCMC steps, normalized by n .

difference of the means to judge whether the means of $\log \mathcal{L}(T)$ within the windows have stabilized. In our experiments, we continuously examine whether the difference falls into the range $[-0.05n, 0.05n]$ to check the equilibrium state, where n is the number of nodes in the network.

In Figure 4-4, we plot the trace of $\log \mathcal{L}(T)$ as a function of the number of MCMC steps, normalized by n . We observe that the Markov chains mix well over all datasets (i.e., $\log \mathcal{L}(T)$ becomes stable soon after the initial state), indicating the convergence to the stationary distributions. Even though the mixing time can be exponential in the worst case [MV05], we observe that, in practice, the Markov chain in *HRG* usually can converge within $1000 \cdot n$ steps on networks of around ten thousand nodes. Figure 4-4 also shows that the integration of differential privacy actually speeds up the movement of the Markov chains and makes them mix even faster. Roughly, the running time of n MCMC steps in our experiments is 0.18s for *polblogs*, 4.1s for *wiki-Vote*, 9.5s for *ca-HepPh* and 22.9s for *ca-AstroPh*. More details about the mixing time can be found

in [CMN07].

Figure 4-4 also shows the comparison of the sampled dendrograms' $\log \mathcal{L}(T)$ in different parameter settings, including that of the dendrogram sampled in the non-private manner. We can observe that, for networks with around ten thousand vertices, $\log \mathcal{L}(T)$ of the dendrogram sampled under a relatively small privacy parameter (e.g., $\epsilon_1 = 0.5$) is still comparable with that under a relatively large privacy parameter (e.g., $\epsilon_1 = 0.9$). Hence, we expect that even assigning a relatively small ϵ_1 for sampling the dendrogram will not significantly harm the data utility of the released network. To validate this, we further conduct experiments with small $\epsilon_1 \in \{0.3, 0.5\}$ and various $\epsilon_2 \in \{0.005, 0.01, 0.1, 0.5, 0.9\}$. The performance shown in Figure 4-9—4-16 confirms that our scheme preserves reasonably good data utility even under a stringent privacy parameter.

4.5.3 Utility Analysis

To show the utility of the released networks, we compare their degree distributions, shortest path length distributions and influential node ranking with those of the original networks. Due to the randomness of our algorithm, we examine the variance of its performance by running the algorithm multiple times on each network for each parameter setting. We observe that the variance in all cases is small. Hence, we randomly pick one graph generated for each dataset and report here.

Degree Distribution. Figure 4-5 shows the degree distributions of the released data under different sanitization schemes, with y -axis in log-scale. It can be seen that, in all cases, *HRG* preserves well the right-skewness of the original networks, meaning that it preserves good distance scale between “hubs” (i.e., nodes having high degrees) and the majority of low-degree nodes.

Shortest Path Length Distribution. Figure 4-6 depicts the shortest path length distribution of each network. We observe that, in general, the released networks preserve the shapes of the distributions with respect to those of the original networks. However, we also observe the increase of paths of small lengths (e.g., 1-3). We believe this is due to the extra edges added to the low levels of the sampled dendrogram, which corresponds to the local structures in a network. But this does not have a big influence on the network's global structure.

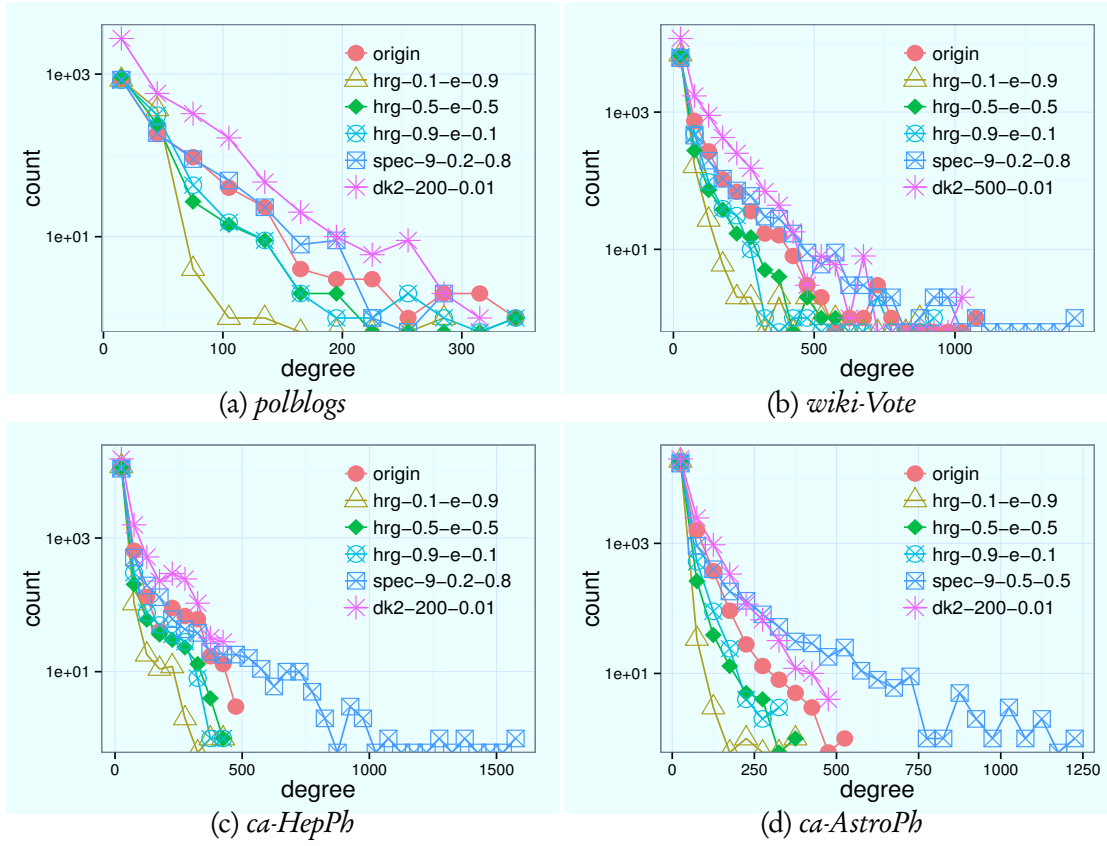


Figure 4-5: Degree distribution

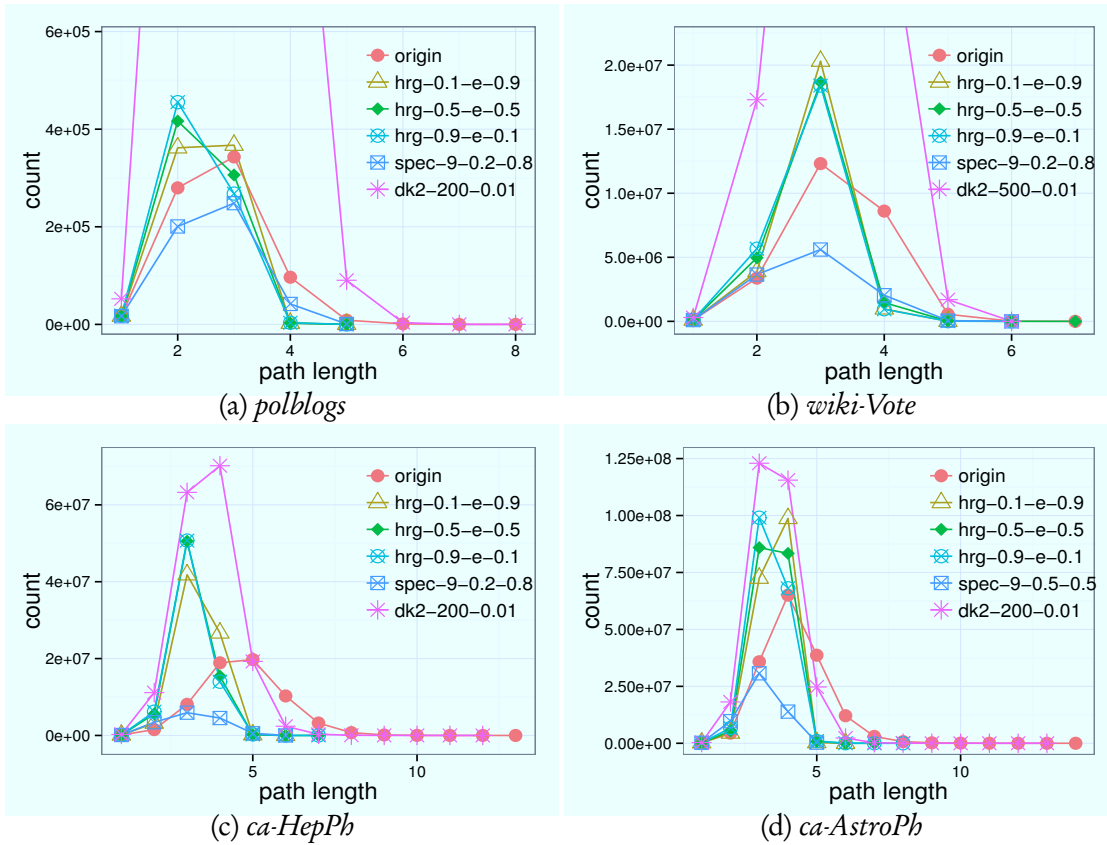


Figure 4-6: Shortest path length distribution

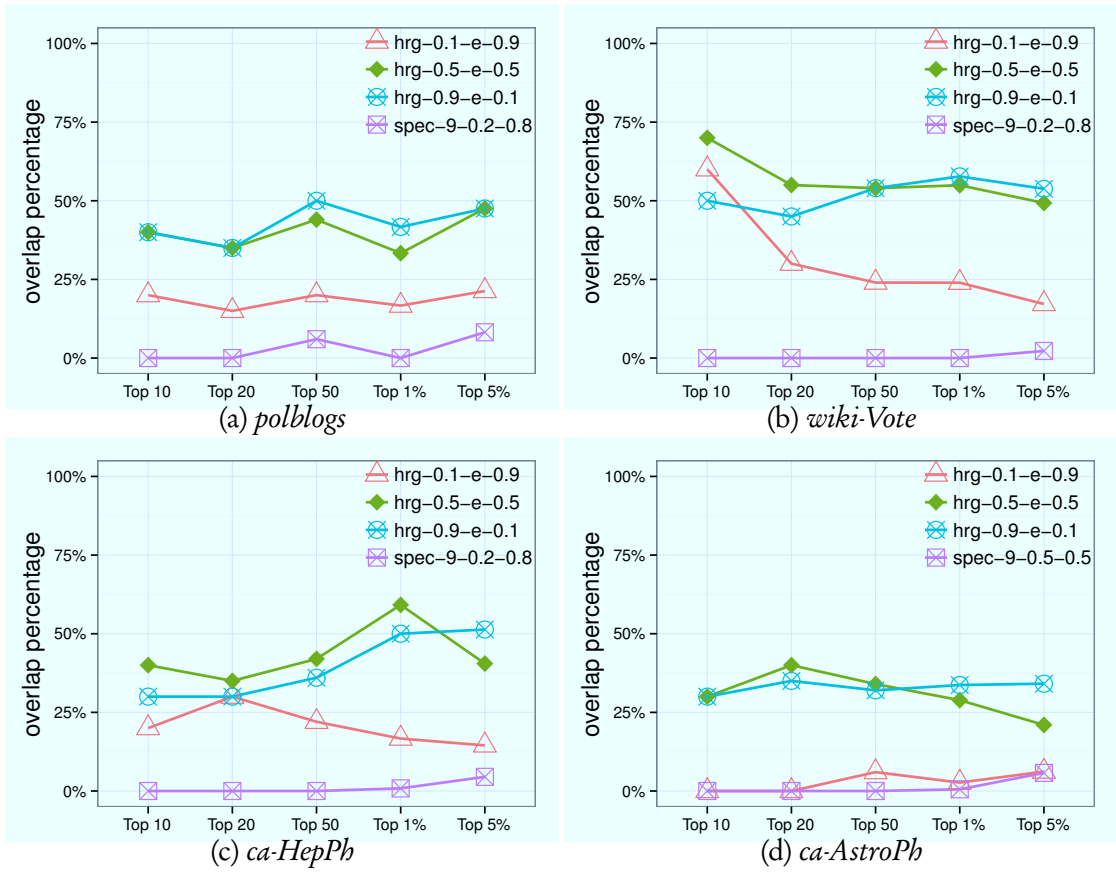


Figure 4-7: Overlaps of top- k vertices

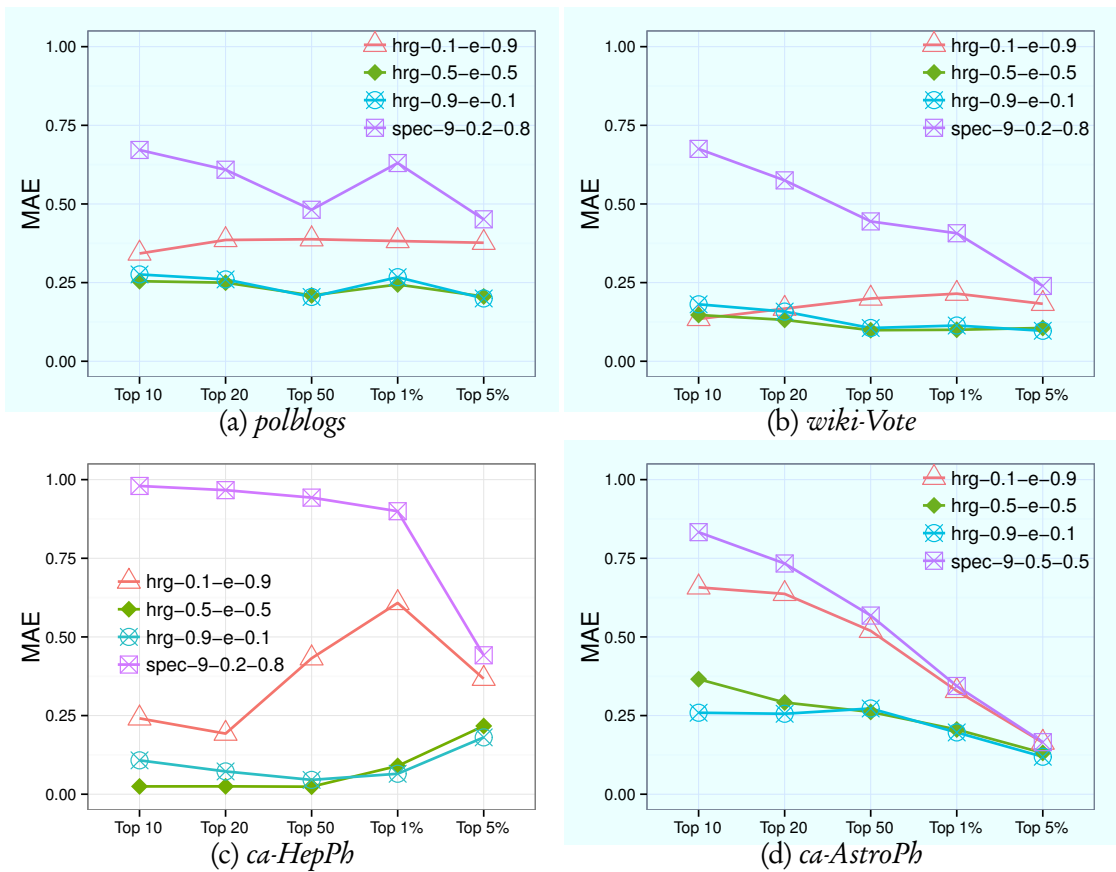


Figure 4-8: Mean absolute error of top- k vertices

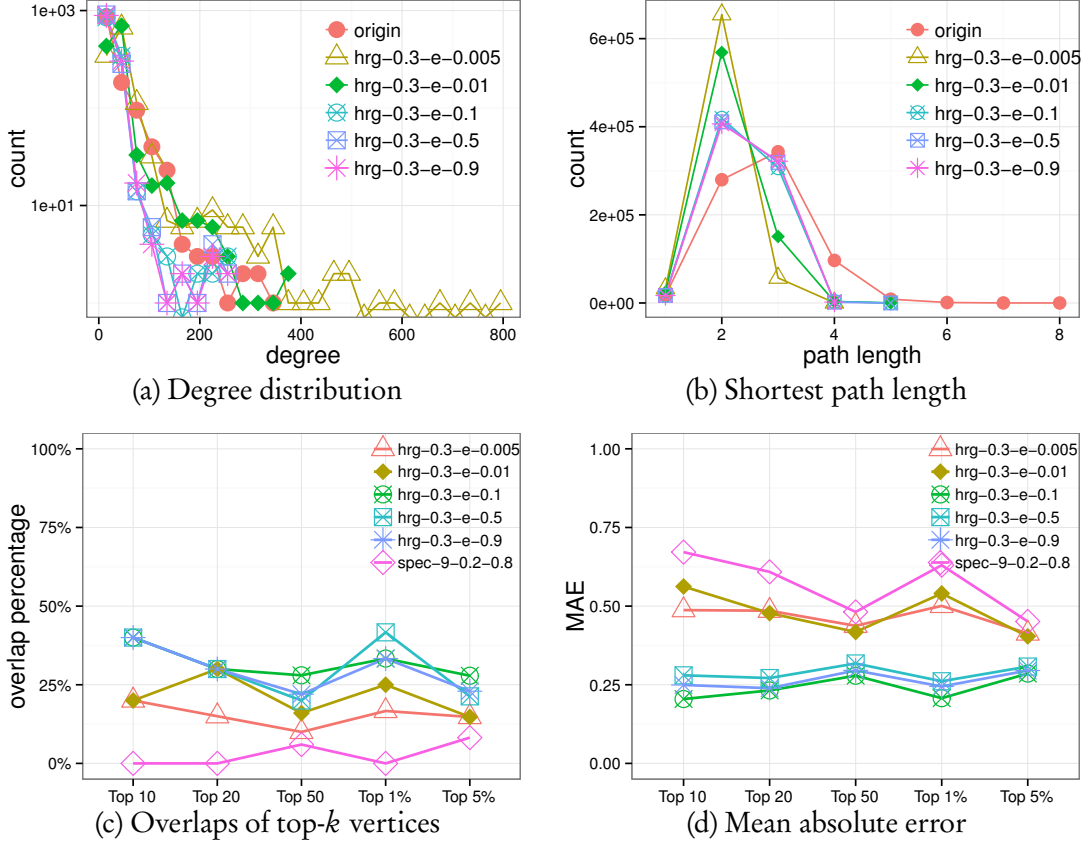


Figure 4-9: *polblogs* with *hrg-0.3*

Influential Node Analysis. In this part we use the same metrics and approaches adopted in Chapter 3’s counterpart. In our experiments, we first test the percentage of common nodes in top- k most influential nodes of the original graphs and those of the sanitized graphs. We examine top 10, 20, 50 influential nodes as well as top 1% and 5% nodes in the networks. The results are presented in Figure 4-7. We see that *HRG* guarantees a consistent 25%-75% overlap of common nodes in all the cases.

Similarly, we then calculate the mean absolute error of the top- k most influential nodes’ EVC scores. Let the set of top- k nodes in the original graph be α and that of the sanitized graph be β . To show that nodes in β have similar centralities to those in α , we use the *mean absolute error* (MAE) to compare the EVC scores in β with those in α . Formally, the MAE value is formulated by: $MAE(\alpha, \beta) = \frac{1}{k} \sum_{i=1}^k |EVC(v_{\alpha}^i) - EVC(v_{\beta}^i)|$, where v_{α}^i and v_{β}^i are the top- i nodes in α and β , respectively. In Figure 4-8, we observe that the MAE of *HRG* with $\epsilon_1 = 0.5$ and 0.9 is reasonably low (e.g., less than 25% in most cases). The overlaps in top- k nodes and the low MAE together indicate that *HRG* well preserves the hub nodes, which represent

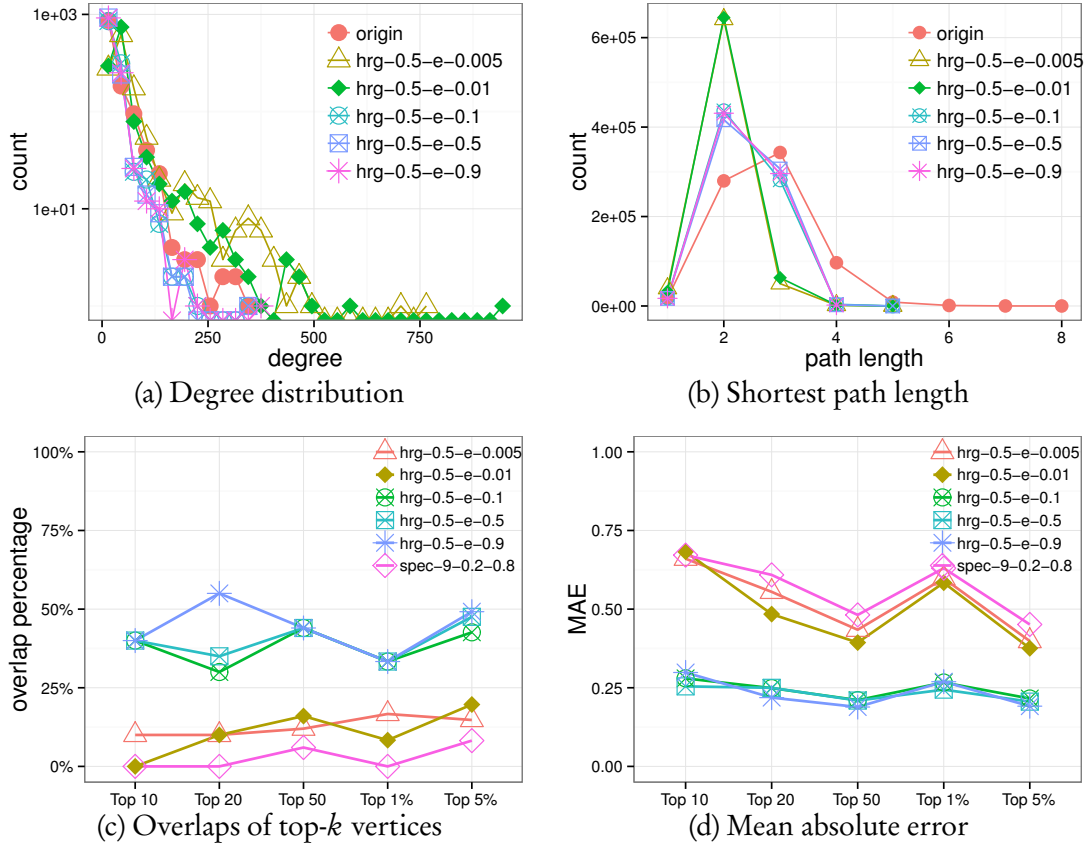


Figure 4-10: *polblogs* with *hrg-0.5*

the global structure of the sanitized graph.

4.6 Summary

In this chapter, we have addressed the privacy concerns in network data release by proposing a novel data sanitization method under differential privacy. Our solution is based on structural inference over the hierarchical random graph (HRG) model. Compared with the existing works, we theoretically prove that the sensitivity of our solution is much smaller, only logarithmic in the order of the network size (i.e., the number of vertices), implying a significant utility improvement. Extensive experiments on four real-life datasets confirm that our solution outperforms the state-of-the-art competitors.

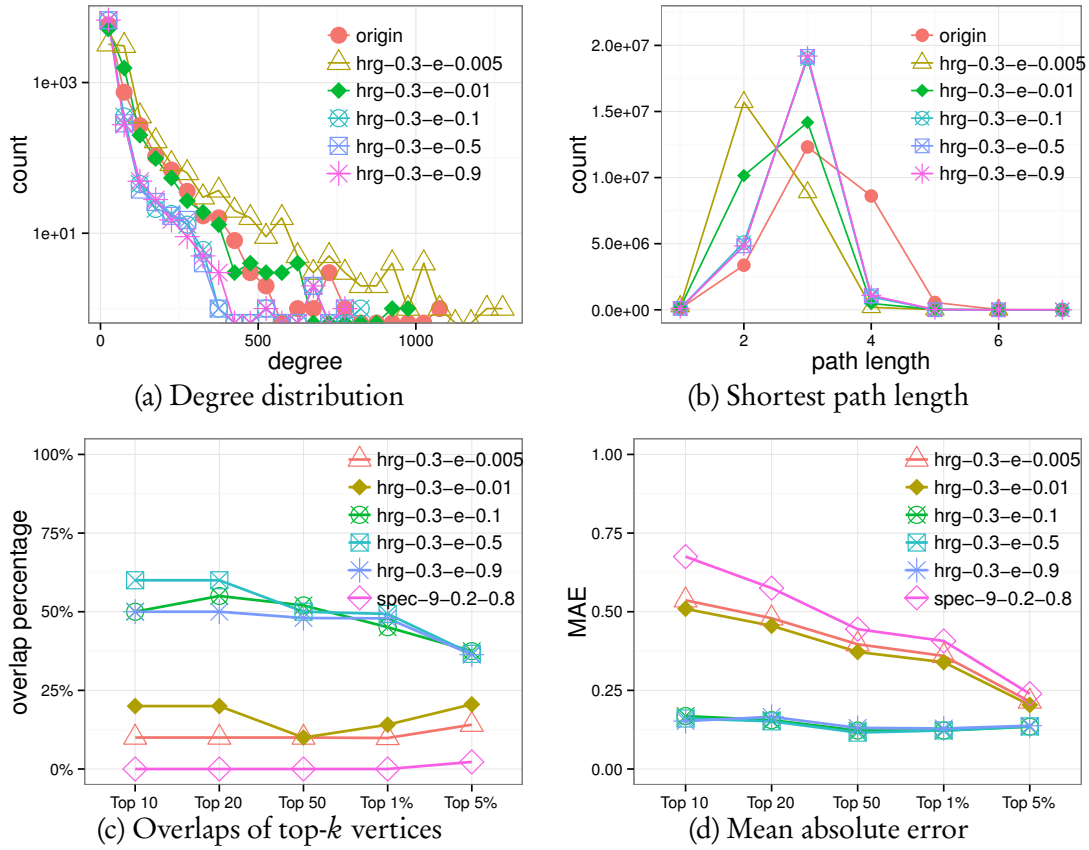


Figure 4-11: *wiki-Vote* with *hrg-0.3*

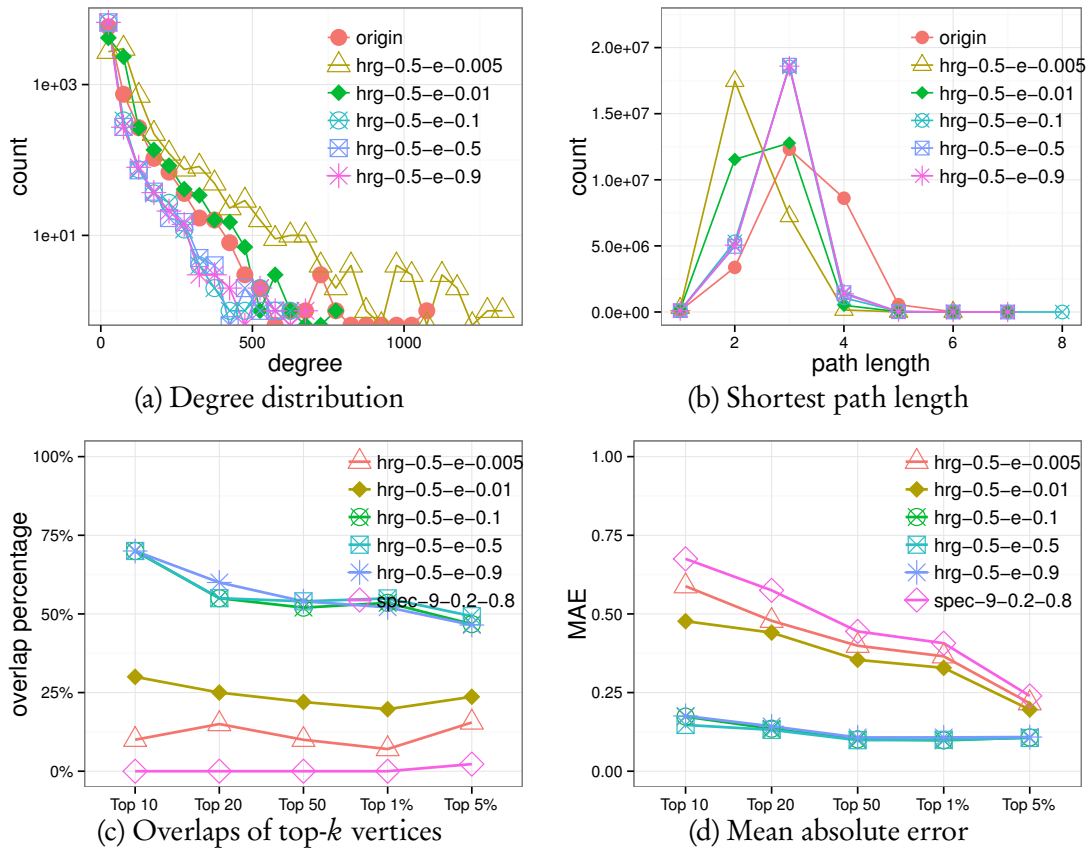


Figure 4-12: *wiki-Vote* with *hrg-0.5*

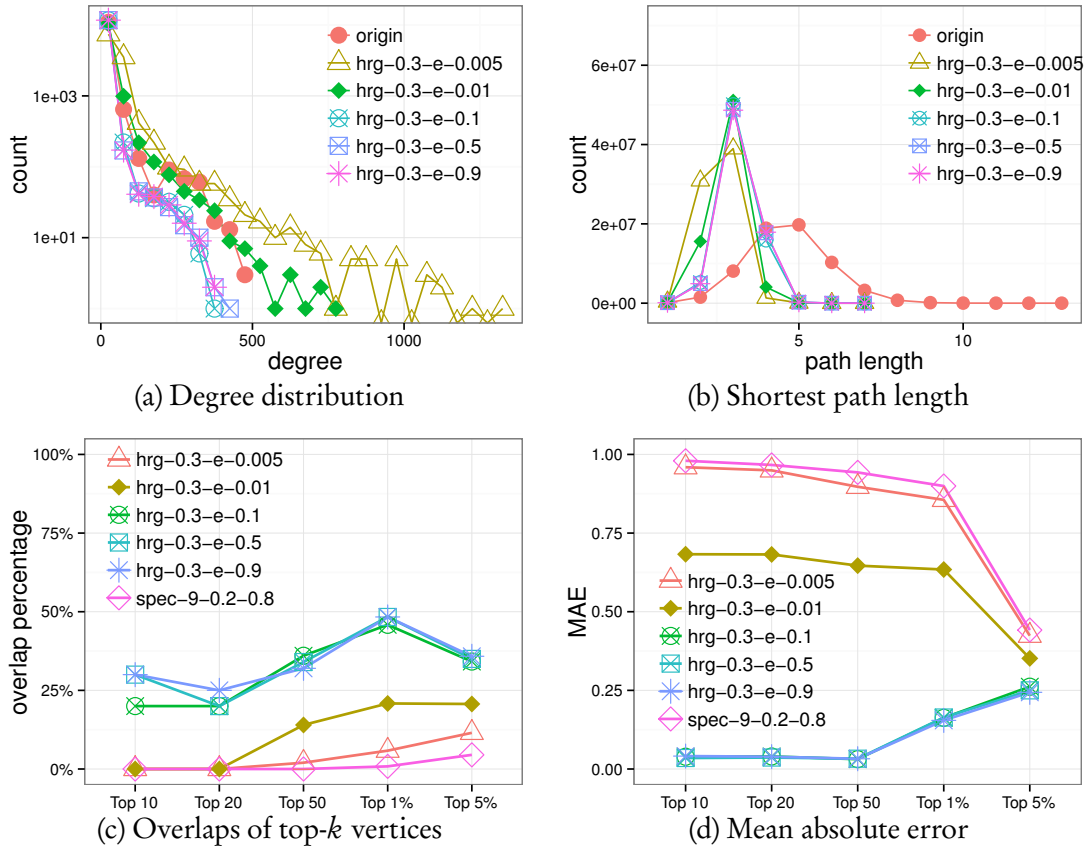


Figure 4-13: *ca-HepPh* with *hrg-0.3*

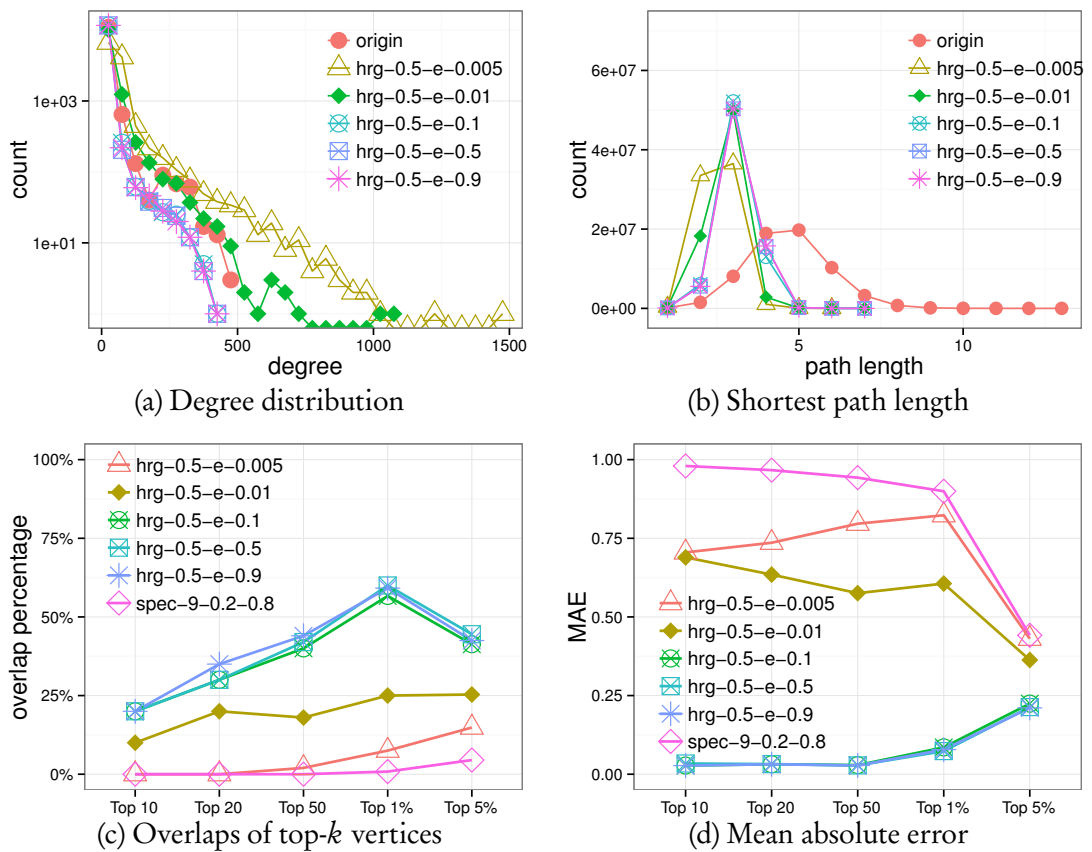


Figure 4-14: *ca-HepPh* with *hrg-0.5*

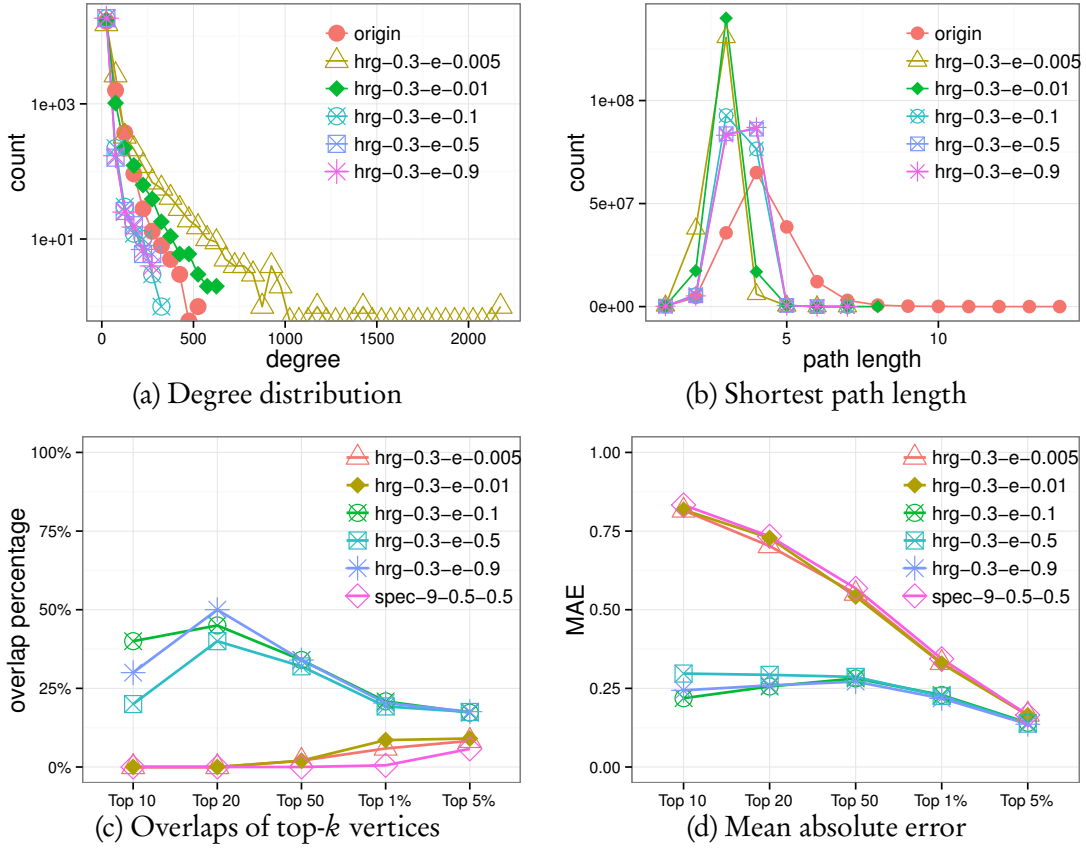


Figure 4-15: *ca-AstroPh* with *hrg-0.3*

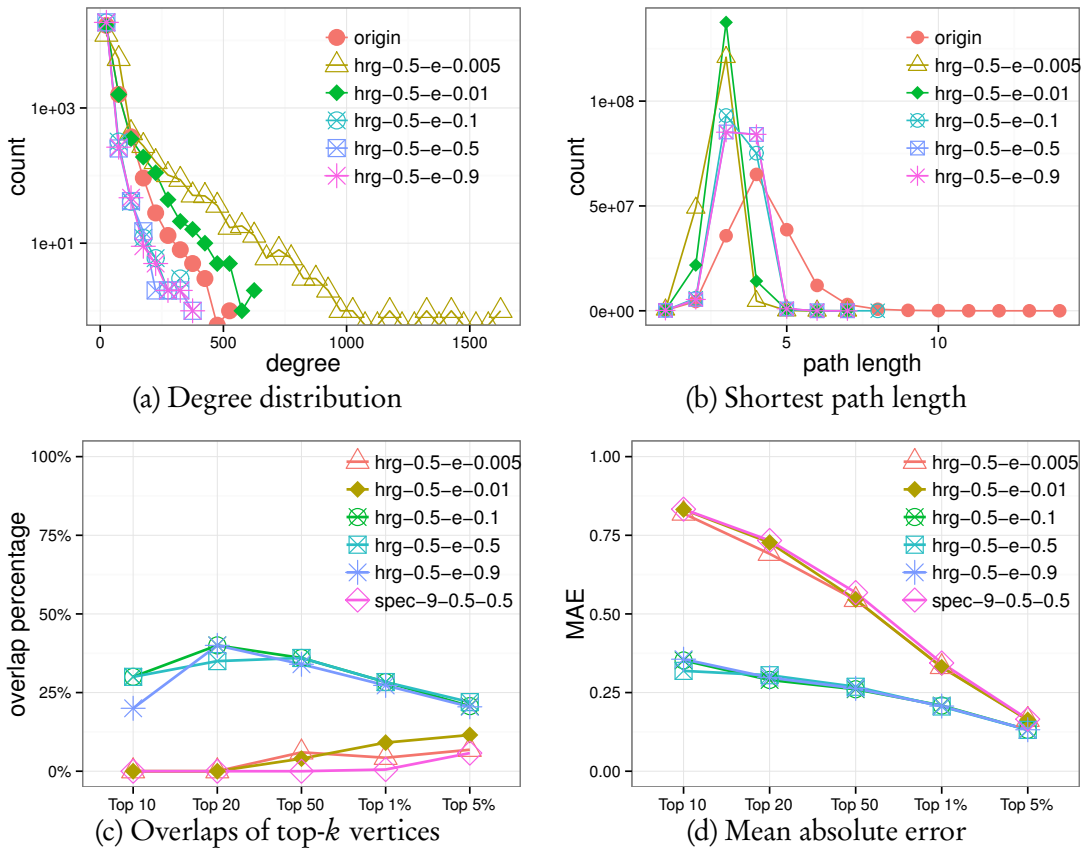


Figure 4-16: *ca-AstroPh* with *hrg-0.5*

Chapter 5

Background and Related Works of OSN Collaborative Access Control

In this chapter, we turn to look at the problem of OSN collaborative access control. We first illustrate a few crucial shifts from traditional access control requirements to today's new challenges that arise in social networks. We also briefly review the state-of-the-art solutions, and meanwhile delineate the scope of our third work.

5.1 Enforcing Access Control in the Social Era

In today's social era, information sharing has become a norm. In particular, OSNs such as Facebook, Twitter, LinkedIn, have dramatically expanded and expedited our access to information. Accompanying such data explosion, OSN users have become creators and managers of their own data. Hence, they are expected to take the responsibility for designing detailed rules to control the ways their data shall be exposed to different groups of audience. In this chapter, we first look at how OSNs shape the massive shifts in access control enforcement. In particular, we illustrate the emerging challenges in enforcing access control in OSNs, which traditional access control mechanisms (e.g., Mandatory Access Control (MAC) and Role-Based Access Control (RBAC)) are unable to support.

5.1.1 Towards large personal-level access control

OSN connects us together. Forming, maintaining and growing social bonds become the key features that OSN supports. Being connected is easy; maintaining the connections is, however, rather complicated. Today, it is common for an ordinary OSN users to have hundreds of friends today. Studies have shown that we share a common connection pattern, i.e., the 5-15-50-150-500 pattern, representing the emotional support group, sympathy group, semi-regular communication group, stable social connection group and weak-tied tangential relationship group, respectively [Ada12]. Clearly, maintaining such a large number of friends can be overwhelming even for sophisticated OSN users.

5.1.2 Towards distance-based and context-aware access control

Compared with traditional RBAC, a crucial difference in OSNs is that, the distance counts. Consider the relationships from friends to friends of friends, and then public. In this case, the trust level reduces as the distance increases. Hence, the design of access control enforcement also needs to factor in the social distance.

Besides, OSN users are often not certain of the desired level of information sharing when they connect to new friends. It is also difficult to ask users to articulate every particular social connections in OSNs. This is because we trust different friends on different topics at different levels. For example, one may expose his or her tweets to all friends when seeking information, but only feel comfortable to express sentiments to a few who are closest. It is evident that the various desired levels of sharing in terms of different contents heavily influence our choice of audience for each particular post.

5.1.3 Towards relationship-composable access control

Unlike traditional RBAC, access control models in OSNs also need to consider how relationships can be composed, that is, $R_1 \circ R_2$. For example, one might consult sensitive health matters who are “professionals in health care” but not “friends” ($R_1 - R_2$), share commercial promotions with those who are both “colleagues” and “fashion lover” ($R_1 \cap R_2$), or boast of career success to “friends” or “friends of friends” ($R_1 \cup R_2$). Today one can easily leverage advanced search engines like Facebook Graph Search to

process queries that are arbitrary combined over social networks. However, this may also lead to social embarrassment or even personal insults if the access control settings are not properly specified ¹.

5.1.4 Towards more collective access control

Being socialable today is not just about connecting to friends. It also has been encouraged by many traditional online activities. For example, Facebook supports tagging people to share group photos; Github views itself more as a social coding community, rather than just a code repository, by encouraging its users to folk others and team up to build software; Mendeley, the online reference manager, facilitates socialization by supporting collaborative groups features, which pushes forward socialization within academic communities. Furthermore, in times of emergency like earthquakes or terrorism attacks, where being connected matters most, information such as messages, photos, notifications and ongoing events is often contributed and shared by many people. Hence, there is clearly an increasing demand to design agile collective access control strategies for such ad-hoc applications and organizations.

5.1.5 Towards more negotiable access control

Another feature of OSNs is its flexibility. Hence it is not suitable to build hard-wired rules in OSN access control systems like in traditional MAC systems. Instead, soft rules shall be supported. For example, in the above scenario, for a group photo, friends may adjust their access control requirements(e.g., public or private) by considering others' privacy preferences and sentimental needs. A collaborative research/coding group can also change their access control strategy at different stages of its development. Hence, a more negotiable access control strategy shall be supported.

Clearly, enforcing access control in OSNs is challenging. Taming such access control management challenges at a personal level can be very difficult for ordinary OSN users. Hence, there is an urgent need to develop new access control frameworks to deal with such challenges.

¹<http://actualfacebookgraphsearches.tumblr.com/>

5.2 State-of-the-art OSN Access Control Strategies

Traditionally, OSNs have largely empowered the publisher of the content, say Alice, to be solely responsible for regulating access to shared content. As such, the research centers on trust management problems, that is, to use trust level to determine the set of users who can have access to the data. Some models in this line [FAZ09] rely on the topology of the social networks to estimate trust levels. Existing OSN platforms like Facebook also adopt such a strategy, allowing users to restrict access regarding *friends*, *friends of friends* or *public*². Some others use the relationships between the publishers and the accessors [Fon11]. In addition, there are also models [CFP06; KGG+06] that consider user reputation to infer the trustworthiness.

Yet another strategy for developing privacy-enhanced social networks is to re-design OSN communication architecture, instead of fully trusting OSN providers. For example, Jahid et al. [JMB11] consider shifting access control enforcement from OSN providers to the users by means of encryption. Another approach is to develop semi or fully decentralized systems [CFP09; JNM+12]. However, it should be pointed out that these solutions have not been widely adopted by any OSNs in practice so far.

The last line of works focuses on leveraging data management to mitigate users' burden on manually specifying OSN privacy settings. For instant, XACCESS [WSL12] is such an automated access control policy specification tool, which employs a hybrid mining method to infer a user's "social roles" regarding his historical activities as well as network topology. More recently, Park et al. [PKK+14] point out that users may not be clear about the optimal states of data sharing levels where they will be comfortable with. They propose a framework which defines different states of shared data, i.e., optimal, under-shared, over-shared, and hybrid states. They also provide approaches to identify each user's actual state and help users parameterize the decision-making process model to set up an optimal sharing level.

In our third work, we particularly focus on the problem of designing collaborative frameworks to support multi-party data sharing for OSNs. In the literatures, there are already quite a number of works along this line [HAJ11; SMJ10; CF11; HAJ12; HAZ+14]. Squicciarini et al. [SMJ10] employ the *Clarke Tax* algorithm [Cla71] as a voting strategy in their proposed model. The *Clarke Tax* strategy disincentivizes

²<http://www.facebook.com/policy.php/>

players to lie about the true valuation of their preference, and thus promotes truthfulness among users. However, the *Clarke Tax* voting strategy is vulnerable to bidder collusion [Wei99]. The small number of players can collude and over express their preference to some extent. In this way they can bend the entire collective decision without paying the clarke tax. Besides, the final decision can be determined directly by only one “pivotal” individual, the one who is willing to pay more (e.g. tax, credit), whereas others with little resources just cannot afford to influence the collective decision.

Carminati et al. [CF11] introduce an enhanced topology-based access control architecture by user collaboration. They also exploited semantic web technologies to support flexible representations of collaborator’s relationships and resources. Hu et al. [HAJ12] formalize a multiparty access control model to address the same issue. Their proposed conflict resolution mechanism aggregates each player’s decision policy and sensitivity towards a specific accessor and thus leverages each player’s preference in collective decision-making. In their very recent work [HAZ+14], they further utilize a game theory strategy to solve this problem where the users all aim to maximize their own benefits. However, in this thesis, we point out that in real-world scenarios, OSN users tend to behave in a more friendly and constructive manner, rather than a selfish way of maximizing personal benefits. It is common that OSN users are vastly influenced by their neighboring friends. Thus, our approach seeks to simulate such positive social interaction by taking into account such peer effects automatically.

Chapter 6

Peer-aware Collaborative Access Control

6.1 Introduction

Many Online Social Networks (OSNs) now offer users free storage to upload their photos¹ online. In addition, these OSNs also provide tools for users to edit photos, stitch photos together, and even make slideshows and galleries. Besides, OSNs also allow users to tag persons in the photo. Tagging a person not only facilitates users to organize the photos, but also encourages photo-sharing in OSNs. For example, the Picasa Web Albums², which has recently been integrated with Google+³, will give the person being tagged permission to view the photo and share with others.

However, if it is not properly managed, photo tagging may violate a person's privacy and lead to his embarrassment. This is because the person being tagged can further share the photo with others. Consequently, the original uploader will lose control over who can access the photo as it may become available for the entire Web to view or be disseminated via Google+ stream. In fact, many inadvertent users may not even be aware of the potential audience's size as they tag people and share their photos with others. Although a user can detag himself from a photo, he cannot stop other tagged users from sharing it in their social networks.

¹In this work, for ease of presentation, we use photo as a shared content. Our method works for other shared content such as video and documents when the co-owners can be identified successfully.

²<http://picasaweb.google.com/>

³<https://plus.google.com>

The widespread concerns to protect user privacy have prompted OSNs to develop access control mechanisms [CF10]. These are largely designed based on relationships and topology of the social networks [facebook; FAZ09; CFP06]. Unfortunately, the decision for regulating the access to the shared photo still rests solely on the uploader of the photo. As such, these access control mechanisms are unable to deal with the privacy concerns of other persons that may appear in the photo.

Intuitively, we can view all persons appearing in a photo as co-owners of the photo. Each of these co-owners can thus voice his opinion about who can have access to the photo. By developing a method that considers the privacy of all co-owners, a collective decision on the access restrictions may be determined. However, everyone has his desired preference of sharing at the appropriate exposure level that he is most comfortable with. It is thus not uncommon that conflicts will arise as a result of differing privacy preferences - while one may be excited about sharing his photo, another may prefer to keep it from public view. How to resolve such conflicts in differing privacy concerns and to support a fair collective decision-making strategy is an open problem.

We also notice that even though the users are very concerned about the privacy, they seldom do much to protect their privacy. In fact, users are just reluctant to spend time in specifying privacy policy. Thus, a practical OSN access control tool should be intuitive, light-weight, and automatic (i.e., require minimal human intervention/effort).

To this end, many researchers recently began to introduce collaborative access control policy-making mechanisms in OSNs [SMJ10; CF11; HAJ12]. These methods integrated the social relationship types and the topology of social networks in the policy-making, as well as in assessing the trust level of accessors. Simple voting functions (e.g. full-consensus, one-override, majority) are provided to deal with privacy conflicts. Moreover, the *intensity* of the user's perceived importance towards a specific preference also matters. For instance, Alice is essentially neutral and do not have any preference on whether to keep the photo private or share with the public; on the other hand, John may be very passionate (and hence has a higher level of intensity than Alice) about sharing photos to the public. Thus, intensity shall also be incorporated into the expression of user preference in access control rules. To promote fairness

and truthfulness among users, a more sophisticated voting method was proposed in [SMJ10] to remove the incentive to conceal the true perceived intensity of a preference.

However, a thoughtful strategy should not only collect each individual's own intention, but also take into account the social interaction among the co-owners in the social network. Co-owners of a photo are typically not business competitors where they need to hide their true intention and compete with one another to achieve maximum gain. Instead, they are likely to be friends/acquaintances/colleagues and hence there is a tendency to be considerate and sensitive to the feelings of one another. Consider the scenario where two close friends, Alice and Bob, had taken a photo together. Initially Bob wants to share this photo with other friends, whereas Alice is strongly against making the photo public. By taking Alice's feeling into consideration, Bob is likely to respect her and change his mind, hence achieving the consensus to keep this photo private. It is inevitable that peers exert tremendous influence on individual behaviors, let alone the ubiquitous interactions on OSNs. Such *peer effects* can be found in a vast literature in the field of sociology and psychology (e.g. [Goy07; Jac08; JV10; TK59; BDF09]).

By taking into account peer effects in making collaborative access control rules, some conflicts of co-owners' intention will disappear naturally. We aim to treat everyone's preference with equal importance so that no single person's personal preference directly dominates the collective decision. At the same time, our proposed strategy, called CAPE, incorporates peer effects, allowing users to adjust their intention according to their neighbors' actions. The goal is to try to achieve more agreements, or even better, full consensus and satisfy everyone's privacy concerns. This is inherently different from collusion which has a negative connotation. In fact, considering peer effects on network may undermine colluding behavior. This is the case as a subgroup of users' decisions do not necessarily directly dominate the entire group's decision. The result of our strategy depends on the overall network structure of peer effects, that is, how each individual reacts to his neighbors.

In this work, we employ a game theoretic model to simulate the continuous decision adjustments that occur in social interactions. The theoretic model guarantees a unique equilibrium under appropriate parameter setting, which ensures the mediation will terminate. In the model, each player (which is a co-owner) expresses his

own preference and his perceived peer effects independently. Each personal setting is private (i.e. no other players know his setting) and will be managed by the central strategy mechanism engine. Moreover, the model offers a direct solution, a “payoff-maximum” action, for each user, automatically. Thus, everyone will be satisfied with the action chosen by such a procedure. At the same time, except the initial set-up, we free the users from any effort and time during the mediation process.

The rest of this work is organized as follows. In the next section, we introduce some preliminaries. In Section 6.3, we discuss the challenges in designing the game theoretic collaborative access control model, and give a big picture of our solution. Section 6.4 presents the setup phase for players (aka co-owners), and Section 6.5 shows the mediation procedure. In Section 6.6, we discuss several related issues. Lastly, we summarize our work in Section 6.8.

6.2 Representation of OSNs

In this section, we introduce the representation of an OSN. Roughly, we can categorize OSNs into two types: distance-based network and circle-based network. The former classifies the users based on the topological distance. For example, hop 1 corresponds to Friends, hop 2 to Friends of Friends and hop $+\infty$ to Public. The latter focuses on the specific classification of an individual’s friends as groups, say Family, Colleague, School-mates. We first illustrate our work on distance-based network. We will discuss how our work can be extended to circle-based network in Section 6.6.3. Now, we shall introduce the core parts of a distance-based OSN and a few notations used in our proposed collaborative access control framework as follows:

- U . The set of OSN users. Assume that each user $u_i \in U$ has a unique id, i .
- E . The set of edges that connects the users. An edge $e \in E$ connects two users, which can be either undirected or directed.
- d_{ij} . Distance from u_i to u_j , which can be measured with the path length.
- Originator. The user who initiates the collaboration. In OSNs the originator is the user who first uploads the photo in his web album for sharing and tags other users who also appear in the photo.

- Co-owner. The user who appears and has been tagged in the photo.
- Player. The user, either the originator or the co-owner, who participates in the collaboration to make collective access control rules. We use the player i and the player u_i interchangeably in this work when no ambiguity arises.
- Access Control Policy Choice Set, \mathbb{C} . On distanced-base OSNs, we can represent a control policy $c \in \mathbb{C}$ in terms of d . For instance, $d = 0$ indicates keeping the photo private, whereas $d = +\infty$ means to share the photo with the public. As in existing work [SMJ10], in this work, for distance-based OSNs, we consider a total of four options, namely, private ($d = 0$), friends ($d = 1$), friends-of-friends ($d = 2$) and public ($d = +\infty$).

6.3 The Big Picture

A straightforward method to make a collective decision in real life is to let each player explicitly express his preference first. After viewing other’s actions, the players can further revise their preference settings [JV10; TK59; BDF09], disclose their new decisions, and so forth until a common decision is reached after a few rounds. This is a typical scenario, as studied in a vast literature on sociology, where the behavior of an individual, say Alice, may change as she is being influenced by her peers.

Now, in our context, we can expect peer effects to come into play too. Among friends/colleagues, there will always be some who are more highly regarded and respected (or even feared); and opinions of such persons are likely to have a greater impact on others’ decisions. For example, Alice may become more inclined to keep a photo private as a result of (some of) her neighbors’ (aka friends and seniors) preferences to keep it private; on the other hand, the change in Alice’s decision may have an impact on others for which she has influence over. Thus, we need a tractable formulation to incorporate such continuous interactions between the peers.

However, in real-life, OSN users often access the network independently and hence not all users will be online at the same time. Thus it is not practical and desirable for any access control mechanisms to require synchronization in time. In addition, the players may be stuck in an endless task since individuals can always adjust their decisions. To this end, we propose a method that simulates the negotiation and interaction

among players, while, at the same time, ensures the simulation will terminate under an appropriate set-up. Specifically, we suggest a mediation procedure that facilitates the following features:

1. Each individual, say Alice, can perform an initial set-up independently (of course, the settings can be updated whenever it is necessary). Essentially, Alice assigns weights to her neighbors to reflect the degree of influence in which her neighbors have over her decision. There is no synchronization required in mediation process, as long as Alice sets up the initial configuration. In addition, Alice does not need to be personally involved in the mediation process, freeing her from the burden of mediation and saving her time.
2. The method should allow Alice to always choose the action that benefits her emotion most. In other words, after considering both her personal willingness and the peer effects, the method should always take the most appropriate strategy from Alice's perspective. We refer to this action as the maximum "emotional payoff" action.
3. The strategy should guarantee a unique *Nash equilibrium*. That is, briefly, the game should always reach a scenario where no player has the incentive to change only his own decision.

As we shall see, our proposed method ensures the above features, provided that each player should not regret the choice he has made in response to the actions taken by other players.

We are now ready to give an overview of our proposed framework - the CAPE framework that facilitates Collaborative Access control by considering *Peer Effects*. Our CAPE framework is depicted in Figure 6-1. The mediation engine, which is the key component, requires input from several sources:

- **OSN structure.** The subgraph of the OSN that involves the co-owners/players.
- **Originator.** The originator triggers the mediation process by uploading the photo and tagging the players.
- **Player.** For each player, two types of information are provided. The first is content-dependent, i.e., his inclination (which we refer to as *intensity* towards

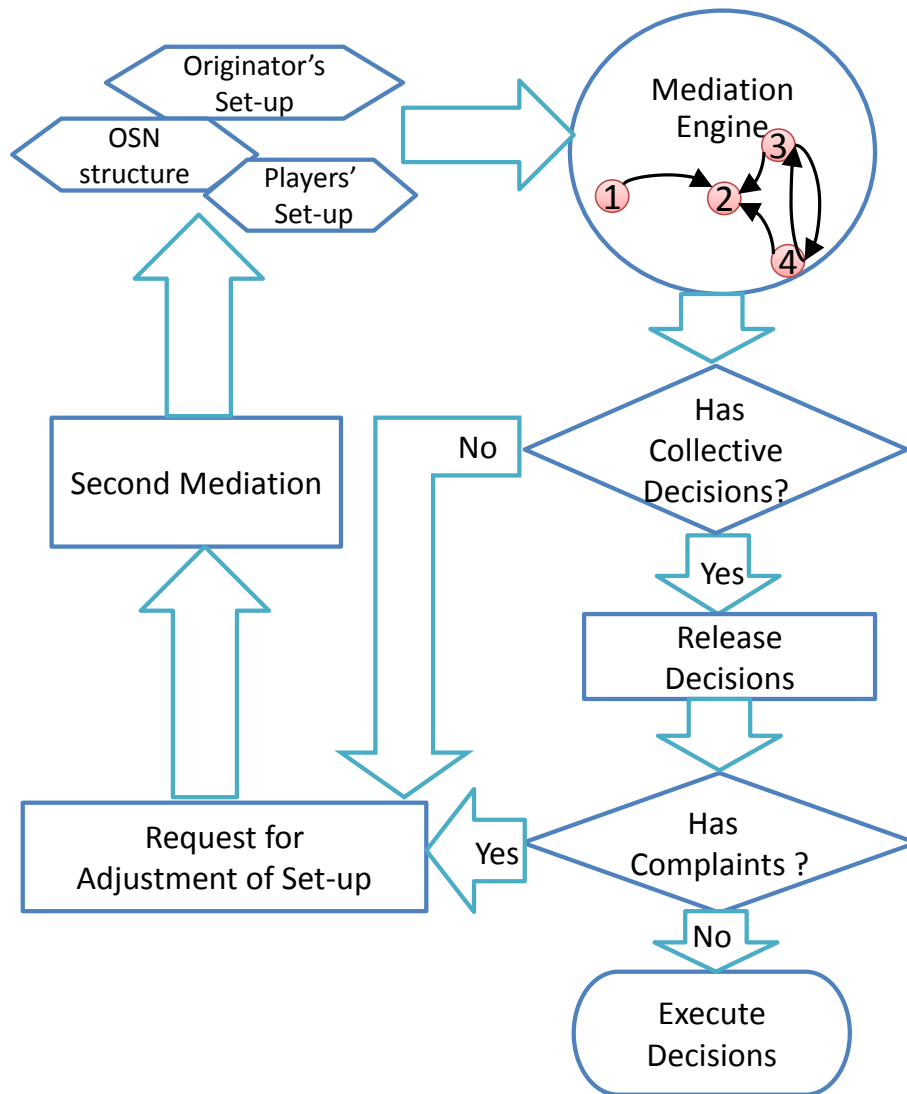


Figure 6-1: The CAPE Framework

the access control policies; this may be different for different shared content, e.g., Alice may be fine with making public an ordinary group photo, while she may not want to share the photo where she felt she may be embarrassed (e.g., she was drunk and was throwing out).

The second is peer-effects-based, which specifies the player's inclination to be influenced by his immediate neighbors. We refer to this inclination as the *peer effects*. This information is more stable. It can be specified once during set-up, and only updated when necessary.

In this work, the mediation engine only considers how a player is influenced by his direct neighbors; moreover, the specification of the degree of influence must be positive. This is because our work models the situation of a constructive environment where players mutually support, or reinforce each other. We shall discuss this further in Section 6.4.

To initiate the mediation process, the originator uploads a photo, and invites the players (other users appearing in the photo) by tagging them. Based on the the social network structure and the players' specification, the mediation starts to simulate the continuous interaction between the players. The mediation process is done for each possible choice independently. In fact, for each choice, an unique equilibrium exists as long as the set-up's configuration meets the required conditions, and then the system can automatically compute the final intensity each player would like to select on the given choice. Once the final intensity of all the choices are ready, the choice with the highest intensity is considered to be the final choice the player would like to select. As each player's final decision is ready, the system will try to make a collective decision with a *voting* function. All the players will be informed of the outcome if any. The whole procedure terminates if the mediation succeeds and no complain arises. In the event where there is no equilibrium, and/or some players find the collection decision unacceptable, the players may have to adjust their inclinations for another round of mediation.

In this work, since the whole procedure aims to achieve more agreements, we suggest two voting functions:

- (i) **full consensus.** Mediation succeeds only when all the players agree on the final decision;
- (ii) **strong majority with a threshold θ .** No fewer than θ percentage of players agree on the final decision.

But for ease of discussion, we only consider full consensus as the voting function in all the examples. We shall discuss the mediation process and the voting function in Section 6.5.

6.4 Player Setup

In our CAPE framework, we need each player to specify his preferences for the available choices, as well as the degree at which he may be influenced by his immediate peers' decisions. We capture these with two types of variables:

- **Intensity Score (I-Score)** $x_i(c_k)$, which measures the inclination/extent to which the player i is willing to take the choice c_k . Its value is unbounded and non-negative. However, to make it more intuitive for the users, in the initial setup, we restrict $x_i^0(c_k)$ to be an integer and let $x_i^0(c_k)$ be between 0 and 5. This essentially corresponds to six attitudes: {strongly disagree, disagree, slightly disagree, slightly agree, agree, strongly agree}. In this way we let all the players specify the values on the same ground. During the mediation peer effects cause $x_i(c_k)$ to change (increase) iteratively. So $x_i(c_k)$ is no longer bounded within the range $[0, 5]$. But the player is only required to assign the initial values of his own I-Scores, the vector \mathbf{x}_i^0 for all the choices.
- **Peer effects Score (PE-Score)** w_{ij} , which characterizes how much weight the player i intends to place on player j 's action. In this work, we require $0 \leq w_{ij} \leq 1$. This ensures the model is a game of strategic complements where each player mutually supports one another.

6.4.1 Setting I-Score

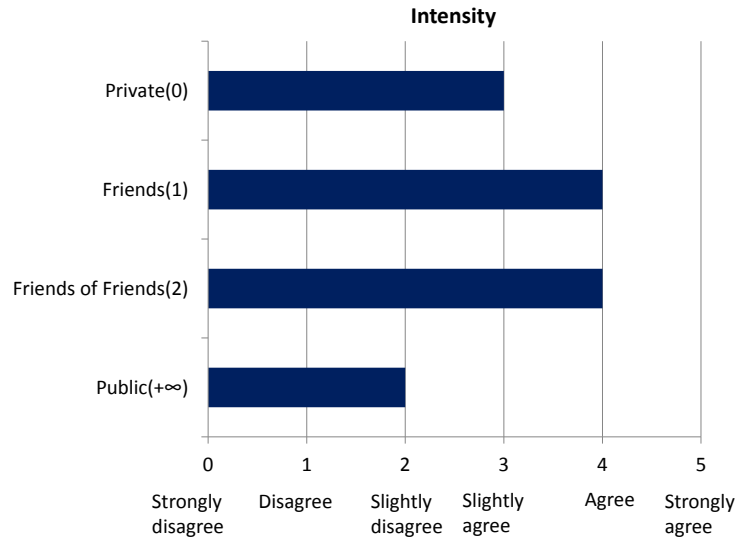
To make an intuitive interface for users, we recommend presenting players the slide bars as tools for specifying the values. We propose two ways to let a user set his I-Scores, as pictured in Figure 6-2a and Figure 6-2b. The first one considers four independent choices: private, friends, friends of friends, public, where $d = 0, 1, 2, +\infty$ respectively. This method allows players to set their I-Scores arbitrarily, i.e., players can set any values for the weights on different choices. A higher intensity score indicates the player has a stronger desire to take the choice. Referring to the setting in Figure 6-2a, the player essentially says: I don't quite agree to share the photo with the world; I sort of prefer to keep it private; however, I would most agree if we restrict the access to just friends or friends-of-friends. In the case where a player set all the I-Scores to be of the same value, we assume the player is willing to undertake any of such actions, regardless

of their initial values (i.e., a player cannot reject all the possible actions together). As we shall see shortly, during the mediation process, each of these choices/options will be considered independently. For ease of reference, we shall refer to this method as **Method OO** (for ‘option only’).

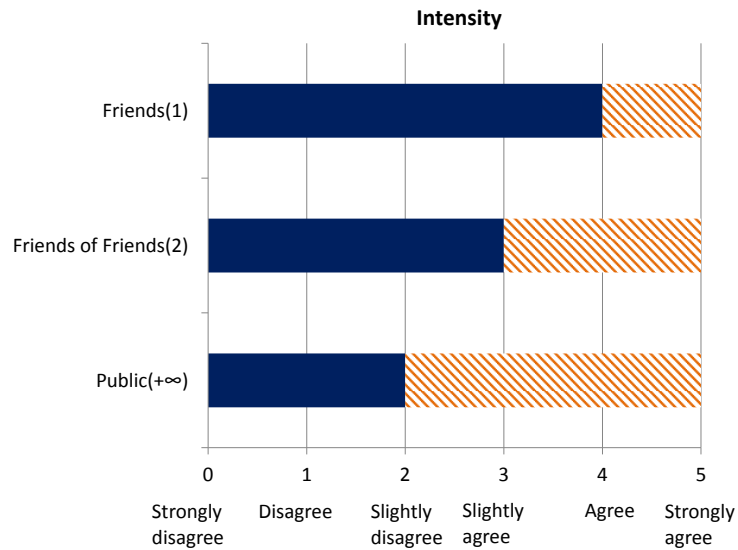
Alternatively, we can design the bars as in Figure 6-2b, where it shows complementary assessment of “Take the choice or not”. As depicted in Figure 6-2b, we consider all the choices except “private”. Given a specific choice c , we consider two actions in turns, “Take c ” and “Against c ”. For example, the length of the solid-color(blue) bar on the top indicates the intensity of picking the choice of “Friend(1)”, whereas the striped(red) bar beside shows the intensity of being against the choice “Friend(1)”. If a player is against all the three choices, it implicitly indicates he would like to keep the photo private. We shall discuss in Section 6.5 how these two complementary actions are independently considered during the mediation process. In other words, we actually have three pairs of choices to work with in the mediation process. We shall refer to this method as **Method OC** (for ‘option and its complement’).

Before leaving this section, we note that the flexibility of arbitrary setting has both pros and cons. On one hand, it allows a player, say Alice, to specify her preferences. For example, a player can specify a value of 5 for “friends” and 0 for all the other options, indicating that he only want to restrict to friends, and nothing else. However, the flexibility may also lead to some undesirable settings, e.g., it does not seem to make sense to have a setting of 5 for “friends-of-friends” and 0 for all other choices. Such flexibility requires users to fully appreciate the consequences of their settings (in order to ensure the settings are meaningful). Moreover, such flexibility also makes it more challenging for all users to reach a consensus.

To achieve more agreements easily, we can enforce some constraints on the setting. One reasonable approach is to assume that, if a player i set $x_i^0(c_k)$ as the I-Score of a choice c_k , then the I-Scores of other more restricted preferences, except “private”, must not be less than x . For instance, if a player slightly agrees with the choice “friends of friends”, he must at least also slightly agree with the more restricted choices like “friends only”. Such a constraint is reasonable since, if one already agrees on a relatively relaxed choice, he cannot be against more private choices. This strategy is akin to asking players to conform to the group by sacrificing the joy of sharing and



(a) Option Only(OO)



(b) Option and its Complement(OC)

Figure 6-2: Two Designs of Intensity Bar

encouraging them to protect privacy.

6.4.2 Setting PE-Score

We note that the PE-Score needs to be set only once. For each player, he essentially maintains an array of the PE-Score for each of his neighbors. All players could have set the PE-Scores for their friends when they first include them as friends. In the event that a player did not provide enough information, the default setting is assumed to be 0, i.e., he will not support others' options.

Table 6.1: Initial I-Scores with Method OO

	Private	Friends	Friends of Friends	Public	Intention
u_1	2	3	3	<u>4</u>	Public
u_2	<u>5</u>	0	0	0	Private
u_3	2	<u>4</u>	2	2	Friends
u_4	<u>5</u>	4	1	1	Private

6.5 The Mediation Process

We now describe the mediation process which is an iterative procedure to simulate the social interaction among the players. To see our approach in action, let us first illustrate the whole procedure with a running example, and then present the proposed mediation mechanism.

6.5.1 An Example

Consider a scenario where Player u_1 is a friend of Player u_2 , and Player u_2, u_3, u_4 are colleagues. Figure 6-3a shows the social network of their relationships in a graph model. Suppose the four of them have taken a photo together. The originator u_1 posted this photo on his own web album, and also wanted to share it on OSNs. So u_1 tagged all other users on this photo, trying to make a collective decision on whether this photo should be posted for public view or be kept private. Since we have only discussed the “full consensus” voting function, we shall illustrate the mediation process with this. Further, let us assume that each player has assigned the I-scores and the PE-Scores. For ease of presentation, we assume users specify their initial I-Scores using **Method OO**, i.e., each player specifies his preference for each option. We defer the discussion when the second method, **Method OC**, is used in Section 6.5.3. Table 6.1 and 6.2 show the players’ assigned values. The value underlined in Table 6.1 corresponds to the action the player prefers most in the beginning.

As it turns out, player 2 is strongly concerned about his privacy over this photo than the others, and he is not going to change his mind according to others’ intention. In contrast, player 1 is more willing to put the photo online; and meanwhile, he also values player 2’s feeling and/or opinion.

From the setting, we observe that conflicts exist in the initial intention among the players. But through the mediation process, as we shall see later, we can derive each

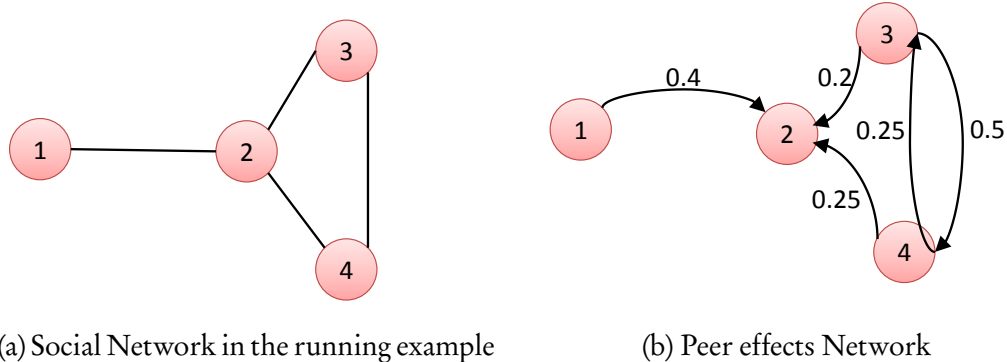


Figure 6-3: Peer effects in OSN

Table 6.2: Peer Effects Scores

	Player 1	Player 2	Player 3	Player 4
Player 1	0	0.4	0	0
Player 2	0	0	0	0
Player 3	0	0.2	0	0.5
Player 4	0	0.25	0.25	0

player’s final intensity scores and their final action when the continuous interaction terminates, as shown in Table 6.3. We note that u_1 turns out to have the same score (of 4) for both Private and Public. As such, we can pick either option. By selecting “Private”, conflicts can be resolved, i.e., through the mediation, a full consensus on keeping this photo private has been reached.

Table 6.3: I-Scores at Equilibrium with Method OO

	Private	Friends	Friends of Friends	Public	Intention
u_1	<u>4</u>	3	3	<u>4</u>	Public/Private
u_2	<u>5</u>	0	0	0	Private
u_3	<u>7</u>	6.86	2.86	2.86	Private
u_4	<u>8</u>	5.71	1.71	1.71	Private

6.5.2 The Mediation Engine

We shall now present the mediation engine used in our work to deal with conflicts that may arise in initial settings. As mentioned, our mediation factors in the peer effects of players. Our scheme is based on a game model, a variation of the game model of Ballester, Calvó-Armengol, and Zenou [BCAZ06], which is also discussed in Chapter 9 of Jackson’s book [Jac08]. In this model, we use the variable *payoff* to describe

Algorithm 6.1: ComputeEquilibIScore

Input : initial I-Score matrix X^0 , PE-Score matrix W , Choice set \mathbb{C} of size n

Output: the I-Score matrix at equilibrium X^{eq}

```
1  $\mu_1(W) \leftarrow$  the largest eigenvalue of  $W$ ;  
2 if  $\mu_1(W) < 1$  then  
3    $X^{eq} \leftarrow X^0$ ;  
4   foreach choice  $c_k \in \mathbb{C}$  do  
5      $\alpha_k \leftarrow$  the  $k^{\text{th}}$  column in  $X^0$ , initial I-Score vector regarding the choice  
6      $c_k$ ;  
7      $I \leftarrow$  the identity matrix;  
8      $\mathbf{x}^{eq}(c_k) \leftarrow (I - W)^{-1} \alpha_k$ ;  
9     let  $\mathbf{x}^{eq}(c_k)$  be the  $k^{\text{th}}$  column of  $X^{eq}$ ;  
9   end  
10  return  $X^{eq}$ ;  
11 else  
12 | return NULL;  
13 end
```

to what extent the player considers a specific adjustment of his I-Score is appropriate in response to the actions of his neighbors. That is, the higher this “emotional” payoff is, the more the player assesses the appropriateness of this adjustment of I-Score. The variable *payoff* p_i of the player i is defined as follows:

$$p_i(c_k) = a_i x_i(c_k) - \frac{b_i}{2} (x_i(c_k))^2 + \sum_{j \neq i} b_j w_{ij} x_i(c_k) x_j(c_k), \quad (6.1)$$

where $a_i \geq 0$ and $b_i \geq 0$ are scalars, and w_{ij} is the *PE-Score* value specified by player i . The expression $-\frac{b_i}{2} (x_i(c_k))^2$ is a force to draw back to player i 's own decision. It is easy to see that a high intensity of the player i 's own intention will tend to inhibit the increase of the payoff. Therefore, player i can see some trade-off by taking further action to adjust his I-Score. Moreover, since $w_{ij} \geq 0$, the payoff tends to increase by considering other's I-Score, $x_j(c_k)$, which thus simulates the interaction where the players reinforce each other's actions. That is, when the intensity of the neighbors' action is high, the intensity of the player's corresponding action would also be high. Intuitively, it describes the phenomenon where an individual tends to conform to the patterns of his peers' behaviors.

We assume that each player always chooses the action that offers the highest “emotional payoff”, and he can never regret the action he takes at each step. Then, we can derive such action to adjust I-Score by setting the derivative of the payoff $p_i(c_k)$ to 0. Hence, such payoff-maximizing action can be described by

$$x_i(c_k) = \frac{a_i}{b_i} + \sum_{j \neq i} w_{ij} x_j(c_k) \quad (6.2)$$

Equation (6.2) indicates that player i should continuously adjust his I-Score regarding other player’s updated I-Score $x_j(c_k)$ in each round. But, in fact, no further user intervention is needed any more. This is because the final I-Scores at equilibrium state can be directly derived with an analytical method. Therefore, we can directly compute the final action for each player based on just the initial setting. To illustrate such analytical solution, let us first denote $\mathbf{x}^{eq}(c_k)$ as the vector solution of such I-Score $x_i(c_k)$ at the equilibrium. And let α_k be the vector of $\frac{a_i}{b_i}$ regarding the choice c_k . Then the vector solution can be expressed as follows,

$$\mathbf{x}^{eq}(c_k) = (I - W)^{-1} \alpha_k, \quad (6.3)$$

where I is the identity matrix. Since b_i is a scalar, we can set $b = b_i = 1$ for all i . Correspondingly, α_k is set to be a_i . W is the matrix whose entry is the PE-Score, w_{ij} . We can further think of W as a weighted and directed network, **the peer effects network** as depicted in Figure 6-3b, where the weight of edge is assigned to be w_{ij} .

In Equation (6.3), the matrix $(I - W)^{-1}$ serves as the factor of peer effects, applied on the vector α_k . At the very beginning, without factoring peer effects, we can consider α_k to be just the vector containing all the initial I-Score regarding the choice c_k . Thus, we can set the entry of α_k , a_i , to be $x_i^0(c_k)$.

With Equation (6.3), we can compute the final I-Scores directly. Algorithm 6.1 describes the entire procedure. The above solution holds if $I - W$ is invertible and $(I - W)^{-1}$ is nonnegative. Ballester et al. [BCAZ06] shows that these conditions can be met if and only if $\mu_1(W) < 1$, where $\mu_1(W)$ is the largest eigenvalue of W . Another sufficient condition to satisfy the conditions is to let all $w_{ij} \geq 0$, and the sum of the entries of each row/column of W be less than 1.

With the aboved formula, we can automatically compute the intensity scores of all

Algorithm 6.2: ComputeDecisions (Method OO)

Input : Players set U , I-Score matrix at equilibrium X^{eq} , Choice set \mathbb{C} of size n , voting function f

Output: the collaborative decision Ω

```
1  $\gamma = [ ]$ ;  
2 foreach player  $u_i \in U$  do  
3    $\mathbf{x}_i^{eq} \leftarrow$  the  $i^{\text{th}}$  row vector in  $X^{eq}$ ;  
4    $\gamma_i \leftarrow \{c_k | x_i^{eq}(c_k) \text{ is the maximum element in } \mathbf{x}_i^{eq}\}$ ;  
5    $\gamma \leftarrow \gamma$  with  $\gamma_i$  appended;  
6 end  
7  $\Omega \leftarrow f(\gamma)$ ;  
8 return  $\Omega$ ;
```

the choices for each player as the mediation reaches equilibrium. The choice with the highest score will be selected as the final action that the player would like to undertake. Formally, the final decision γ_i of player i is,

$$\gamma_i = \arg \max_{c_k \in \mathbb{C}} x_i(c_k) \quad (6.4)$$

In **Method OO**, we compare a player's I-Scores of all the choices together and select the one with the highest score to be his final decision. Algorithm 6.2 illustrates this procedure.

6.5.3 Constraining the I-Score Setting

Now, it is possible that with more choices/options and arbitrary setting of I-Score values, the chances of achieving agreements decreases. A solution to this problem, as described in Section 6.4, is to be “biased” towards privacy by restricting users to always specify equal or greater intensity scores for more restricted choices. To see this explicitly, we use the second method, **Method OC**, to illustrate. Table 6.4 shows an example of user settings satisfying the above constraints. We shall first consider whether all the players agree on the preference “Public”. Recall that under **Method OC**, each choice results in two actions, and they are to be mediated independently; and the final decision for the choice is determined by the action with the larger I-Score. Table 6.5a shows the result of the mediation process for “Public” and “Against

Table 6.4: Initial I-Scores with Method OC

	Friends	Friends of Friends	Public	Intention
u_1	5	5	4	Public
u_2	3	3	0	Friends of Friends
u_3	4	4	0	Friends of Friends
u_4	2	2	2	Private

Table 6.5: I-Scores at Equilibrium with Method OC

	Public	Against Public
u_1	<u>4.0</u>	3.0
u_2	0	<u>5.0</u>
u_3	1.14	<u>9.29</u>
u_4	2.29	<u>6.57</u>

(a)

	Friends of Friends	Against Friends of Friends
u_1	<u>6.2</u>	0.8
u_2	<u>3</u>	2
u_3	<u>6.83</u>	3.6
u_4	<u>4.46</u>	4.4

(b)

Public”. As shown in Table 6.5a, u_1 prefers “Public” while the rest vote for “Against Public”. Since there is no full consensus, we continue to consider the next pair of choices, “Friends of Friends” and “Against Friends of Friends”. This time, as shown in Table 6.5b, all the players agree on the choice “Friends of Friends”. So the mediation stops here and a final decision is reached to sharing this photo within the distance no more than “Friends of Friends”. The algorithmic description of the procedure is given in Algorithm 6.3.

Algorithm 6.3: ComputeDecisions (Method OC)

Input : Players set U , initial I-Score matrix X^0 for each choice type, PE-Score matrix W , Choice type set \mathbb{C} of size n , voting function f

Output: the collaborative decision Ω

```
1  $\Omega \leftarrow \emptyset$ ;  
2 while  $\mathbb{C} \neq \emptyset$  do  
3    $c_k \leftarrow$  the choice with the largest distance  $d$  in  $\mathbb{C}$ ;  
4    $\neg c_k \leftarrow$  the choice that is against  $c_k$ ;  
5    $\mathbb{C} \leftarrow \mathbb{C} - \{c_k\}$ ;  
6    $X^0 \leftarrow$  current I-Score matrix regarding  $\{c_k, \neg c_k\}$ , the choice  $c_k$  and its  
   complement;  
7    $X^{eqI} \leftarrow \text{ComputeEquilibIScore}(X^0, W, \{c_k, \neg c_k\})$ ;  
8   if  $X^{eqI} \neq \text{NULL}$  then  
9      $\gamma = [ ]$ ;  
10    foreach player  $u_i \in U$  do  
11      if  $x_i^{eqI}(c_k) > x_i^{eqI}(\neg c_k)$  then  
12         $\gamma_i \leftarrow \{c_k\}$ ;  
13      else if  $x_i^{eqI}(c_k) < x_i^{eqI}(\neg c_k)$  then  
14         $\gamma_i \leftarrow \{\neg c_k\}$ ;  
15      else  
16         $\gamma_i \leftarrow \{c_k, \neg c_k\}$ ;  
17      end  
18       $\gamma \leftarrow \gamma$  with  $\gamma_i$  appended;  
19    end  
20     $\Omega \leftarrow f(\gamma)$ ;  
21  end  
22  if  $\Omega \neq \emptyset$  then  
23    break;  
24  end  
25 end  
26 return  $\Omega$ ;
```

6.6 Discussion

In order for our CAPE framework to be developed into a full-fledge robust solution for practical use, there are several issues that need to be addressed. Here, we shall focus on three of them: (a) How to guide the players to configure the set-up; (b) How to facilitate second mediation in order to achieve more agreements; and (c) how to extend our work to circle-based OSNs. We will discuss each of these in the following subsections.

6.6.1 Configuring the set-up

One of the key parameters in the CAPE framework is the setting of the players' PE-Scores, i.e., how each player views the influence of his peers over his decision. However, the framework is meaningful only if an equilibrium exists. In particular, equilibrium exists and is unique when the PE-Score matrix is not overly dense. The PE-Score matrix is dense when the players over-rely on each other's decision. For example, every player may want the opinion of every other player. As a result of such cross-effect, the intensity of a player's choice is always positively reinforced by other players, which in turn leads to an unbounded increase in the intensity of individual's action (and thus the model cannot reach an equilibrium state).

Therefore, it is important to guide the players to assign or adjust their PE-Score values to ensure an equilibrium state. Overall, a guideline for the players is to assign the PE-Score moderately. In fact, in our framework, we have restricted the PE-Score to direct neighbors, i.e., a player only provide the PE-Score for his immediate neighbors. This helps to reduce the chance for large cross effect (since the PE-Score matrix becomes more sparse). However, even with this restriction, it is still possible that equilibrium cannot be reached. In fact, when the PE-score matrix contains a column, say Col j , and its corresponding row, say Row j such that the sums of the values of Col j , and Row j are both greater or equal to 1, then no equilibrium can be reached. Intuitively, this can happens when player j is very prominent among the other players, and, at the same time, player j also tends to respect his followers' opinions. A solution to handle this case is to let player j dominates his own decision (i.e., ignore other players peer effects), preventing the existence of feedback loops. To see this in action,

Table 6.6: PE-Scores before adjustment

	Prof	s_1	s_2	s_3	s_4
Prof	0	0.25	0.25	0.25	0.25
s_1	1	0	0	0	0
s_2	1	0	0	0	0
s_3	1	0	0	0	0
s_4	1	0	0	0	0

Table 6.7: PE-Scores after adjustment

	Prof	s_1	s_2	s_3	s_4
Prof	0	0.2	0.2	0.2	0.2
s_1	1	0	0	0	0
s_2	1	0	0	0	0
s_3	1	0	0	0	0
s_4	1	0	0	0	0

Table 6.8: Initial I-Scores in the extreme case

	0	1	2	$+\infty$	Decision
Prof	0	<u>4</u>	0	0	Friends
s_1	3	3	3	3	Any
s_2	3	3	3	3	Any
s_3	3	3	3	3	Any
s_4	3	3	3	3	Any

let us consider the following extreme case.

Example 6.1. Consider the scenario where a professor and his four students took a photo together. All of the students respect the professor's opinion. However, at the same time, the professor also decides to conform to his students' choices, as shown in Table 6.6. The mediation cannot proceed due to the very large cross effect between the professor and his students. It can be observed that both Row 1 and Col 1 are greater or equal to 1. $\mu_1(W)$, the current greatest eigenvalue of the PE-Score matrix, is 1, which does not satisfy the condition for existence of the equilibrium. One possible solution to this case is to let the professor reduce his dependency on his students, like making a adjustment as showed in Table 6.7. Table 6.8 and Table 6.9 show the mediation outcome after such adjustment.

We thus develop the following heuristics to address this problem. Given the PE-Score matrix, we determine if the greatest eigenvalue of the PE-Score matrix is less than 1. If so, we expect the existence of an equilibrium. Otherwise, we try to find out a player i such that the sum of the values of row i and column i are greater than

Table 6.9: I-Scores at Equilibrium in the extreme case

	0	1	2	$+\infty$	Decision
Prof	12	<u>32</u>	12	12	Friends
s_1	15	<u>35</u>	15	15	Friends
s_2	15	<u>35</u>	15	15	Friends
s_3	15	<u>35</u>	15	15	Friends
s_4	15	<u>35</u>	15	15	Friends

or equal to 1, and request player i to revise his PE-score. In particular, it has been recommended, as part of the sufficient condition for the existence of equilibrium, that the sum of each player's PE-Score (the sum of each row) should be less than 1, unless he really wants to fully rely on the others' opinions. This process is repeated until an equilibrium can be reached.

6.6.2 Second Round of Mediation

Recall that our goal is to resolve the conflicts that arise in making collaborative decisions. So far, we have assumed that we can always reach a consensus that is acceptable to all players in one round of the mediation procedure. However, the mediation may fail. This happens when full consensus cannot be reached with regard to the set-up. It may also occur when not all players are satisfied with the collaborative decision derived from CAPE with a majority-mode voting function. In this section, we discuss how to further facilitate mutual collaboration if complaints about the outcome arise. The idea here is to identify a key player, say John, who has the highest effect on the aggregate outcome, and let the players who are not satisfied with the outcome turn to John for help. But notice that a unique feature of our method is that the final outcome closely depends on the peer effects network. Collusion is hard to succeed in such circumstance, since an individual or a small group, or even the key player John, does not necessarily dominate the result. Nevertheless, we can still encourage the players to approach John, who gets a lot of respect from his neighbors and has a high overall impact on the group, for help. Because such key player may easily persuade his followers to change their settings as well. In this way, the players can publicly request a second-chance mediation, instead of trying to employ colluding behaviors in private. Since John often gets more respect from his neighbors, it is more likely that the aggregate collective result be reduced optimally if he is willing to change his

Table 6.10: Intercentrality Scores

	Intercentrality
u_1	1.96
u_2	1
u_3	3.8
u_4	2.7

Table 6.11: Adjusted Initial I-Scores with Method OC

	Friends	Friends of Friends	Public	Intention
u_1	5	5	4	Public
u_2	3	3	0	Friends of Friends
u_3	4	2	0	Friends of Friends
u_4	2	2	2	Private

intention score. But we should stress that this does not necessarily lead to a biased aggregate outcome. The outcome is still closely affected by the peer effects network. In addition, we suggest that all the players should have the right to know who the key player is, whether there is anyone who has turned to the key player for help, and whether the key player agrees to adjust his intention or not. It should be a public procedure for petition for another round of mediation, which is distinguished from the colluding behavior (that are done in private).

In Ballester et al.'s work [BCAZ06], they showed that the key player can be identified by ranking the players' *intercentrality*. Let $M = [I - W]^{-1}$, and m_{ij} be its entry. m_{ij} can also be written as $\sum_{k=0}^{+\infty} w_{ij}^k$. This expression counts the number of weighted paths that start from i and end at j . With the matrix M , we define the intercentrality of player u_i as follows:

$$\eta_i = \frac{\left(\sum_{j=1}^n m_{ij}\right)^2}{m_{ii}}$$

The intercentrality actually "counts the total number of direct and indirect weighted paths that hit i " [BCAZ06]. Briefly, it considers not only a player's centrality, but also his contribution to other's centrality.

Example 6.2. Consider a scenario where, using the second method, the player u_4 is not satisfied with the outcome "Friends of Friends" for he is really concerned about privacy. So u_4 asks the originator u_1 for a second mediation, and requests to see who the key player is in their current peer effects network. Table 6.10 shows the intercentrality of each player. As

Table 6.12: I-Scores at Equilibrium with Method OC in the Second Mediation

	Friends of Friends	Against Friends of Friends
u_1	<u>6.2</u>	0.8
u_2	<u>3</u>	2
u_3	4.54	<u>5.88</u>
u_4	3.89	<u>4.97</u>

(a)

	Friends	Against Friends
u_1	<u>6.2</u>	0.8
u_2	<u>3</u>	2
u_3	<u>6.82</u>	3.6
u_4	<u>4.46</u>	4.4

(b)

it turns out, u_3 is the key player. Assume that u_4 talks to u_3 , and persuades u_3 to change his intensity score. u_3 resets his set-up as [“Friends”, 4], [“Friends of Friends, 2”], [“Public”, 0]. All the players are also informed that u_4 is not satisfied with the previous outcome and has asked the key player u_3 to reconsider his setting. If the originator agrees to set up a second mediation, all the players can reset their set-up and then a new mediation begins. Suppose, in this example, all the other players do not change their setting. The new result becomes the choice “Friends”, as shown in Table 6.12b, which further protects the player’s privacy as a result of u_4 ’s complaint.

6.6.3 Circle-based Social Network

In this part, we discuss how to extend our strategy to circle-based social networks. In circle-based social networks, users categorize their friends into different groups. We adapt the method developed by Hu et al [HAJ12]. Essentially, it is not practical to list out all the policy choices by taking into account all the player’s circles together. Instead, we let each player considers the trust level of every accessor from his own perspective. Specifically, given an accessor, each player specifies the intensity in terms of his own circles to decide whether to grant access to this accessor. For example, given a photo, Alice may have the following settings for her circles: (Family, agree), (Lab-mates, slightly agree), (Strangers, disagree). Based on this setting, Alice’s preference for the accessor depends on which circle the accessor belongs to. In our example, if

Alice wants to share the photo with John and John is Alice's labmate, then Alice is essentially saying she is fine with sharing the photo with John. On the other hand, if John is a stranger to Alice, then Alice basically opts to keep the photo private. In some sense, what we really have is an implicit choice which is determined by setting of the circle which the accessor falls into. We can then derive the collaborative result based on each player's intensity score towards this given accessor.

Instead of taking the average aggregate decision as in [HAJ12], our strategy facilitates the players to adjust their decision towards the strangers outside their own circles by considering peer effects. This is based on the assumption that one would like to put more trust on a stranger as this stranger is also a friend of his friends. To see why this is useful, let us consider the following example.

Example 6.3. *Consider the scenario where the player u_i does not know the accessor a personally. However, the accessor a is in fact in the circle of one of u_i 's friend, u_j . As u_i is not familiar with a , u_i cannot accurately assess the risk to share the photo with a . Alternatively, with our strategy, u_i can refer to u_j 's opinion since u_j may know a well. Note that u_i does not need to predict whether a knows his friend u_j or which exact extended circle a belongs to. Since in circle-based networks, one is not likely to know the constituents of other's circles, our method helps the user get a better assessment of a stranger's risk by looking at others' actions.*

6.7 User Interface

We have implemented CAPE as a Facebook application. It is now available at <http://cape-facebook.herokuapp.com>⁴.

CAPE is hosted on Heroku⁵, a cloud application platform. It consists of one PHP application to deploy user interface and connect Facebook API⁶, as well as one backbone Python application with Flask⁷ to realize the functionality of our mediation engine. Currently, we adopt **Method OO** in this implementation of CAPE.

⁴We should stress that because of the frequent changes in Facebook privacy policy, CAPE might need to slightly change the settings or migrate to new versions in the future.

⁵<https://www.heroku.com/>

⁶<https://developers.facebook.com/docs/graph-api>

⁷<http://flask.pocoo.org/>

CAPE

Collaborative Access control by considering Peer Effects

Please enable sharing *My photos to Apps others use* in the Facebook [privacy settings](#) and let your friends to see pictures you are tagged enabling *Who can see posts you have been tagged in on your timeline?* in [Timeline and Tagging settings](#).

This app needs to find pictures where you are tagged together with friends. Without these permissions your friends that are running this app are not allowed to find pictures with you.

You can safely disable these permissions again after your friends collaborate tagging their pictures with you.

Thank you for your collaboration.

Please  Log in with Facebook

Figure 6-4: CAPE-Login

Hi Lorenzo 
 If you are not Lorenzo  please [logout](#)

PE-Score

Select how much you are willing to change your decision to meet your friend idea





		74%
		87%

Figure 6-5: CAPE-PEScores

CAPE requests its user to give the permissions to access the user’s basic information, his photos and photos shared with him on Facebook. Besides, CAPE also requests at least some of the user’s Facebook friends are also using CAPE and can see the posts he has been tagged. It is because CAPE needs his friends’ participation in the procedure, too.

Briefly, CAPE consists of the following steps. In the first step, the originator needs to login with his Facebook account. Figure 6-4 shows the web interface of CAPE to explain the detailed permission issues and request users to login to Facebook.

After login, CAPE lists all the originator’s friends using CAPE, and asks the user to update the PEScores, as shown in Figure 6-5.

Next, CAPE loads the photos with tagged friends who are also using CAPE. The originator now can configure the I-Scores for each photo. A screenshot of this step is taken as shown in Figure 6-6.

Hi Lorenzo [please logout](#)

I-Score

Select how much you agree to share each photo with the reference group from 0 to 5 where:
 0: Strongly Disagree
 1: Disagree
 2: Slightly Disagree
 3: Slightly Agree
 4: Agree
 5: Strongly Agree

[I have had enough for today, let me finish](#)

	Private (0)	<input type="range" value="1"/>	<input type="text" value="1"/>
	Friends (1)	<input type="range" value="3"/>	<input type="text" value="3"/> make default
	Friends of Friends (2)	<input type="range" value="2"/>	<input type="text" value="2"/> make default
	Public (+∞)	<input type="range" value="1"/>	<input type="text" value="1"/>
	Private (0)	<input type="range" value="3"/>	<input type="text" value="3"/>
	Friends (1)	<input type="range" value="5"/>	<input type="text" value="5"/> make default
	Friends of Friends (2)	<input type="range" value="4"/>	<input type="text" value="4"/> make default
	Public (+∞)	<input type="range" value="4"/>	<input type="text" value="4"/>
	Private (0)	<input type="range" value="5"/>	<input type="text" value="5"/>
	Friends (1)	<input type="range" value="1"/>	<input type="text" value="1"/> make default
	Friends of Friends (2)	<input type="range" value="0"/>	<input type="text" value="0"/> make default
	Public (+∞)	<input type="range" value="0"/>	<input type="text" value="0"/>

Figure 6-6: CAPE-I-Scores

Hi Lorenzo [please logout](#)

Evaluate results

According to your and your friends scores, this is the privacy we suggest for your photos.

Please, give us your feedback.

Photo	Suggested privacy	Do you agree?
	friends	<input type="radio"/> Yes <input type="radio"/> No
	friends of friends	<input type="radio"/> Yes <input type="radio"/> No
	public	<input type="radio"/> Yes <input type="radio"/> No
	private	<input type="radio"/> Yes <input type="radio"/> No

Figure 6-7: CAPE-Mediation Outcome

After all the players' configurations are collected, CAPE will present the originator the mediation outcome it derives. Figure 6-7 shows such an example. We should stress that this is an asynchronous procedure. Hence the results may not likely to be available immediately. The user may collect the results next time when he logs in after all the settings have been collected.

6.8 Summary

In this chapter, we have revisited the problem of protecting user privacy in online social networks (OSNs). In particular, we have investigated the design of access control mechanisms for protecting shared content where co-owners may have differing and conflicting privacy preferences. A novel collaborative access control mechanism has been designed. Our key insight is that peer effects should be a key contributing factor to be considered in resolving conflicting preferences. Our proposed framework, CAPE, is based on graph theoretic model, and is able to lead to consensus that is acceptable to the co-owners. Our CAPE framework can be applied to both distance-based and circle-based networks. We have also looked how the peer effects scores should be set to ensure equilibrium. Moreover, we have also discussed how to handle the scenario when a player may not be satisfied with the outcome.

Chapter 7

Conclusion and Future Directions

The goal of research on privacy is to develop mechanisms to protect an individual's privacy and to prevent unauthorized access or leakage of sensitive data. Effective methods will be able to tame public fears of hidden privacy leakage and bring back the trust over the Internet in this digital era. In recent years, large scale integration between e-commerce tech giants and OSNs is clearly on the upswing. The prevalence of OSN apps in app eco-systems also yields an increasing demand to access users' data in OSNs. As such, there is a trend to fuse and integrate data. It is hence very urgent to develop faithful and yet efficient privacy-preserving techniques for OSNs, and to do it rapidly.

This thesis is intended to investigate practical techniques to protect OSN users' privacy. As practitioners, we've covered two topics of privacy-preserving practices, one from the enterprise's point of view and another from that of the individual. In this chapter, we recap the major advances and our contributions on each topic, see how the topics are related in the cutting edge research arena, and point out the main challenges that are emerging in new directions.

7.1 Towards Faithful & Practical Privacy-Preserving OSN data publishing

In our first two works, we've covered two privacy-preserving mechanisms for OSN data publishing, one employing *anonymity* and another using *differential privacy*(DP). We also give a coherent view of the overall development in defining information pri-

vacy, that is, how our understanding in information privacy have changed and matured over the last decade.

Recall that anonymity (including randomization, k -anonymity, l -diversity, etc.) was the first mainstream privacy model adopted by many works. Our first work LORA also falls into this category. LORA considers just to publish simple undirected graphs. However, in real-world scenarios, it is not uncommon to see graphs often contain other additional information. For example, in [SKX+12], we investigate the release of networking data where the edges are labeled. Our method adopts l -diversity as the privacy model. There are also works on graphs that contain weights and directions on edge [SMG+12; DEA12].

As DP now has become the emerging standard for data publishing, many works employ it for answering summary statistics of the underlying data. For example, we have looked at publishing counting summaries on streaming binary data in [CXG+13]. There are also numerous works on publishing histograms, trajectories and frequent items counting problems. However, there are so far limited progress for synthetic data approximation, which is particularly obvious for network data. This in part is because of current DP mechanisms' limitations. But another major reason, we think, is the missing of links between statistics and graph theory. It is still not clear now which summary statistics really capture the entire function of a network.

In our second work, we tend to view the network itself as statistical data, a sample drawn from an underlying distribution. This is particularly meaningful in the real world, since the formation of real-world networks has some elements of randomness. We've shown that our method has significant improved accuracy under the same DP level, compared to other state-of-the-art approaches. The intuition behind our approach is that, by mapping a graph to another statistical model space and sampling in the calibrated statistical distribution, we can effectively control the influence caused by the change in the input. Specifically, we can limit the influence only on one parameter in the model while leaving the rest intact. One interpretation is that, even though the network itself is essentially high dimensional, its intrinsic dimension can be very low in most real-world scenarios. The parameters of the model in a high dimension space are often interlocking, the independent components can be more clearly seen once the graph has been transformed into a low dimensional space. However, the

global sensitivity in DP, if not through careful design, can be easily affected by the high extrinsic dimension/network size(akin to the curse of dimension appearing in machine learning). Hence, it is crucial to first reduce the dimension via sampling, approximating, or mapping graph to other feature domains, in order to lay the ground for constructing the low sensitivity.

As such, we hope our methodology can call out further development of methods in this line of work. It will be interesting to see how more existing sampling or approximation methods on graph can naturally fulfill DP, to avoid directly injecting noises into each part of the feature model. In this way, the impact of the previously “prohibitive” sensitivity that result in poor data utility can be diluted through these sampling or approximation processes.

7.2 Integrating data-access policies with differential privacy

In our third work, we’ve demonstrated a collaborative access control strategy. The main observation is that, in the case where a collective data-access policy is needed, it is common that some OSN users’ decision would be greatly influenced as they consider their peers’ privacy needs. Many works in this line assume, in such scenario, OSN users’ benefits shall be competing with each other. That is, each user tends to selfishly maximize his own gain. However, in contrast, we point out that it is more suitable to assume OSN users tend to be considerate about their friends’ emotional needs. This is more reasonable since OSN users are typically friends. To this end, we’ve designed a framework to simulate emotional negotiation, in which OSN users can adjust their data-access policies regarding such peer effects. We wish our design can function as a knot, providing more flexibility for OSN users in support of constructing a positive, collaborative atmosphere for collective decision-making.

It’s also worth to point out another key feature of our design. That is, our mechanism is also a data-driven model. The final collective decision depends on how each OSN user perceives his friends, in terms of peer effect scores. Clearly, as OSN users become the data creators, many users’ privacy preferences are data-driven and context-aware. Hence, it is also pressing to enable policies to support this change. More

recently, some researchers propose a few works devoted to bridge such data-access policy-making strategies with differential privacy [KM12; HMD14]. The authors advocate to integrate differential privacy with policy-making procedures, by allowing the users to specify *secrets* and *constraints*. The line of works is poised to lead to further development in data-driven access control strategies. We believe it is equally important to develop works in similar spirit for OSN data.

7.3 New privacy issues on emerging applications

In the second part of this thesis, we've also reviewed a variety of solutions for access control enforcement in OSNs. One line of these solutions focuses on controlling the information flow over OSNs, by assuming users shall not trust and rely on OSNs' own protection mechanisms to protect their privacy. However, none of the proposed systems, such as encryption-based and decentralized system, has been widely adopted in real world.

In contrast, OSN users are increasingly dependent on OSNs and third-party developers. This raises more concerns over user privacy. First, many OSNs such as Google+ and Weibo advocate their users to use user-defined circles/groups for OSN content sharing. Hence, OSN users today feel much more protected and comfortable in using OSNs as platforms for sharing content online. While these privacy-preserving mechanisms are more powerful, they are also much more complicated. Clearly, it is not practical to predefine all circles a user will ever need. OSNs also do not currently have an effective mechanism for a user to create and/or customize dynamic (ad-hoc) circles for each publishing session. Hence, more advanced tools for facilitating OSN users to use circles are needed. To this end, we propose in [XAT12] a recommendation framework – the Circle OpeRation RECommendaTion (CORRECT) framework – to assist users in easily utilizing circles and creating ad-hoc circles as needs arise. We believe many more such auxiliary tools are needed to help users better manage the sophisticated privacy settings that today's OSNs provide.

Second, as cloud computing services and mobile apps become prevailing, we've seen an increasing exposure of OSN users' geographic information and transaction records. This prompts the need for protecting these sensitive data compounded with

OSN information, while still allowing users to benefit the convenience brought by these new services. However, in most cases, the users essentially have no control on or have no idea about how their data is used, or whether the usage of their data is reasonable and necessary. In the case of mobile apps, there is a tendency for the apps to ask more permissions to access data than needed. As such, there are some works dedicated to design new data-derived and semantically meaningful disclosure models for relational databases [BKG+13; BKG14]. The goal of these works is to enable strict control over information disclosure while keep them accountable and explainable. We believe it is equally pressing to extend the same line of work on OSNs, since many mobile apps also demand the users' OSN data for their services.

In conclusion, it can be a long-term battle for privacy practitioners to put privacy into practice in OSNs. This is mainly because OSNs is continuously evolving and yielding numerous variant applications in this social era. From another prospective, this also leads practicing privacy in OSNs to be an exciting and enticing research area where much more effort are needed. We hope that, through rich collaborations with many diverse disciplines, we can have further understanding in privacy, and make it truly fulfilled in practice on social networks.

Bibliography

- [Ada12] Paul Adams. *Grouped: How Small Groups of Friends are the Key to Influence on the Social Web*. New Riders, 2012.
- [Agg07] Charu C. Aggarwal. “On Randomization, Public Information and the Curse of Dimensionality.” In: *ICDE*. 2007, pp. 136–145.
- [AMP10] Dino Pedreschi Anna Monreale and Ruggero G. Pensa. “Anonymity Technologies for Privacy-Preserving Data Publishing and Mining”. In: *Privacy-Aware Knowledge Discovery: Novel Applications and New Techniques*. 2010. Chap. 5, pp. 111–141.
- [Bac11] Lars Backstrom. *Anatomy of Facebook*. <https://www.facebook.com/notes/facebook-data-team/anatomy-of-facebook/10150388519243859>. 2011.
- [BDK07] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. “Wherefore Art Thou R3579x?: Anonymized Social Networks, Hidden Patterns, and Structural Steganography”. In: *Proceedings of the 16th International Conference on World Wide Web*. 2007, pp. 181–190.
- [BCAZ06] Coralio Ballester, Antoni Calvó-Armengol, and Yves Zenou. “Who’s Who in Networks. Wanted: The Key Player”. In: *Econometrica* 74.5 (2006), pp. 1403–1417.
- [BJ06] Michael Barbaro and Tom Zeller Jr. *A Face Is Exposed for AOL Searcher*. The New York Times <http://select.nytimes.com/gst/abstract.html?res=F10612FC345B0C7A8CDDA10894DE404482>. 2006.
- [BKG14] Gabriel Bender, Lucja Kot, and Johannes Gehrke. “Explainable Security for Relational Databases”. In: *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*. 2014, pp. 1411–1422.
- [BKG+13] Gabriel M. Bender, Lucja Kot, Johannes Gehrke, and Christoph Koch. “Fine-grained Disclosure Control for App Ecosystems”. In: *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. 2013, pp. 869–880.
- [BCK+09] Smriti Bhagat, Graham Cormode, Balachander Krishnamurthy, and Divesh Srivastava. “Class-based Graph Anonymization for Social Network Data”. In: *Proc. VLDB Endow.* 2.1 (2009), pp. 766–777.
- [BGT11] Francesco Bonchi, Aristides Gionis, and Tamir Tassa. “Identity obfuscation in graphs through the information theoretic lens”. In: *ICDE*. 2011, pp. 924–935.

- [BGT14] Francesco Bonchi, Aristides Gionis, and Tamir Tassa. “Identity obfuscation in graphs through the information theoretic lens”. In: *Information Sciences* 275 (2014), pp. 232–256.
- [BDF09] Yann Bramoullé, Habiba Djebbari, and Bernard Fortin. “Identification of peer effects through social networks”. In: *Journal of Econometrics* 150 (1 2009), pp. 41–55.
- [BGJ+11] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Taylor & Francis, 2011.
- [CT08] Alina Campan and Traian Marius Truta. “A Clustering Approach for Data and Structural Anonymity in Social Networks”. In: *In Privacy, Security, and Trust in KDD Workshop (PinKDD)*. 2008.
- [CXG+13] Jianneng Cao, Qian Xiao, Gabriel Ghinita, Ninghui Li, Elisa Bertino, and Kian-Lee Tan. “Efficient and accurate strategies for differentially-private sliding window queries”. In: *EDBT*. 2013, pp. 191–202.
- [CF10] Barbara Carminati and Elena Ferrari. “Privacy-aware access control in social networks: Issues and solutions”. In: *Advanced Information and Knowledge*. 2010, pp. 181–195.
- [CF11] Barbara Carminati and Elena Ferrari. “Collaborative access control in on-line social networks”. In: *CollaborateCom*. 2011, pp. 231–240.
- [CFP06] Barbara Carminati, Elena Ferrari, and Andrea Perego. “Rule-Based Access Control for Social Networks”. In: 2006, pp. 1734–1744.
- [CFP09] Barbara Carminati, Elena Ferrari, and Andrea Perego. “Enforcing access control in Web-based social networks”. In: *ACM Trans. Inf. Syst. Secur.* 13.1 (2009), 6:1–6:38.
- [CFL10] James Cheng, Ada Wai-chee Fu, and Jia Liu. “K-isomorphism: Privacy Preserving Network Publication Against Structural Attacks”. In: *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*. 2010, pp. 459–470.
- [Cla71] Edward H. Clarke. “Multipart pricing of public goods”. In: *Public Choice* 11 (1971), pp. 17–33.
- [CMN08] Aaron Clauset, Cristopher Moore, and M. E. J. Newman. “Hierarchical structure and the prediction of missing links in networks”. In: *Nature* 453 (2008), pp. 98–101.
- [CMN07] Aaron Clauset, Cristopher Moore, and Mark E. J. Newman. “Structural Inference of Hierarchies in Networks”. In: *Proceedings of the 2006 Conference on Statistical Network Analysis*. 2007, pp. 1–13.
- [CSY+08] Graham Cormode, Divesh Srivastava, Ting Yu, and Qing Zhang. “Anonymizing Bipartite Graph Data Using Safe Groupings”. In: *Proc. VLDB Endow.* 1.1 (2008), pp. 833–844.
- [DEA12] Sudipto Das, Ömer Egecioglu, and Amr El Abbadi. “Anónimos: An LP-Based Approach for Anonymizing Weighted Social Network Graphs.” In: 2012, pp. 590–604.
- [Dem07] Dave Demerjian. *Rise of the Netflix Hackers*. <http://archive.wired.com/science/discoveries/news/2007/03/72963>. 2007.

- [DMN+06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. “Calibrating Noise to Sensitivity in Private Data Analysis”. In: *Proceedings of the Third Conference on Theory of Cryptography*. 2006, pp. 265–284.
- [DP13] Cynthia Dwork and Rebecca Pottenger. “Toward practicing privacy”. In: *JAMIA* 20.1 (2013), pp. 102–108.
- [Fon11] Philip W. L. Fong. “Relationship-based access control: protection model and policy language”. In: *CODASPY*. 2011, pp. 191–202.
- [FAZ09] Philip W. L. Fong, Mohd M. Anwar, and Zhen Zhao. “A Privacy Preservation Model for Facebook-Style Social Network Systems”. In: *ESORICS*. 2009, pp. 303–320.
- [FWC+10] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. “Privacy-preserving Data Publishing: A Survey of Recent Developments”. In: *ACM Comput. Surv.* 42.4 (2010), 14:1–14:53.
- [Goy07] Sanjeev Goyal. *Connections: An Introduction to the Economics of Networks*. Princeton University Press, 2007.
- [HGP09] Sami Hanhijärvi, Gemma C. Garriga, and Kai Puolamäki. “Randomization techniques for graphs”. In: *In Proc. of the 9th SIAM Conference on Data Mining*. 2009.
- [HR12] Moritz Hardt and Aaron Roth. “Beating Randomized Response on Incoherent Matrices”. In: *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*. 2012, pp. 1255–1268.
- [HLM+09] Michael Hay, Chao Li, Gerome Miklau, and David Jensen. “Accurate Estimation of the Degree Distribution of Private Networks”. In: *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining*. 2009, pp. 169–178.
- [HLM+11] Michael Hay, Kun Liu, Gerome Miklau, Jian Pei, and Evimaria Terzi. “Privacy-aware Data Management in Information Networks”. In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. 2011, pp. 1201–1204.
- [HMJ+07] Michael Hay, Gerome Miklau, David Jensen, Philipp Weis, and Siddharth Srivastava. “Anonymizing social networks”. In: *Computer Science Department Faculty Publication Series* (2007), p. 180.
- [HMJ+08] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. “Resisting Structural Re-identification in Anonymized Social Networks”. In: *Proc. VLDB Endow.* 1.1 (2008), pp. 102–114.
- [HMD14] Xi He, Ashwin Machanavajjhala, and Bolin Ding. “Blowfish privacy: tuning privacy-utility trade-offs using policies”. In: *SIGMOD Conference*. 2014, pp. 1447–1458.
- [HAJ11] Hongxin Hu, Gail-Joon Ahn, and Jan Jorgensen. “Detecting and resolving privacy conflicts for collaborative data sharing in online social networks”. In: *ACSAC*. 2011, pp. 103–112.

- [HAJ12] Hongxin Hu, Gail-Joon Ahn, and Jan Jorgensen. “Multiparty Access Control for Online Social Networks: Model and Mechanisms”. In: *IEEE Transactions on Knowledge and Data Engineering* 99. PrePrints (2012).
- [HAZ+14] Hongxin Hu, Gail-Joon Ahn, Ziming Zhao, and Dejun Yang. “Game Theoretic Analysis of Multiparty Access Control in Online Social Networks”. In: *Proceedings of the 19th ACM Symposium on Access Control Models and Technologies*. 2014, pp. 93–102.
- [Jac08] Matthew O. Jackson. *Social and Economic Networks*. Princeton University Press, 2008.
- [JV10] Matthew O. Jackson and Xavier Vives. “Social Networks and Peer Effects: An Introduction”. In: *Journal of the European Economic Association* 8.1 (2010), pp. 1–6.
- [JMB11] Sonia Jahid, Prateek Mittal, and Nikita Borisov. “EASiER: Encryption-based Access Control in Social Networks with Efficient Revocation”. In: *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. 2011, pp. 411–415.
- [JNM+12] Sonia Jahid, Shirin Nilizadeh, Prateek Mittal, Nikita Borisov, and Apu Kapadia. “DECENT: A decentralized architecture for enforcing privacy in online social networks”. In: *PerCom Workshops*. 2012, pp. 326–332.
- [KRS+11] Vishesh Karwa, Sofya Raskhodnikova, Adam Smith, and Grigory Yaroslavtsev. “Private Analysis of Graph Structure”. In: *PVLDB* 4.11 (2011), pp. 1146–1157.
- [KM12] Daniel Kifer and Ashwin Machanavajjhala. “A Rigorous and Customizable Framework for Privacy”. In: *Proceedings of the 31st Symposium on Principles of Database Systems*. 2012, pp. 77–88.
- [KGG+06] Sebastian Ryszard Kruk, Slawomir Grzonkowski, Adam Gzella, Tomasz Woroniecki, and Hee-Chul Choi. “D-FOAF: Distributed Identity Management with Access Rights Delegation”. In: *ASWC*. 2006, pp. 140–154.
- [LLV07] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. “t-Closeness: Privacy Beyond k-Anonymity and l-Diversity”. In: *ICDE*. 2007, pp. 106–115.
- [LT08] Kun Liu and Evimaria Terzi. “Towards Identity Anonymization on Graphs”. In: *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*. 2008, pp. 93–106.
- [MKG+07] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramkrishnan Venkatasubramanian. “L-diversity: Privacy Beyond K-anonymity”. In: *ACM Trans. Knowl. Discov. Data* 1.1 (2007).
- [McS10] Frank McSherry. “Privacy integrated queries: an extensible platform for privacy-preserving data analysis”. In: *Commun. ACM* 53.9 (2010).
- [MT07] Frank McSherry and Kunal Talwar. “Mechanism Design via Differential Privacy”. In: *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*. 2007, pp. 94–103.

- [MV05] Elchanan Mossel and Eric Vigoda. “Phylogenetic MCMC algorithms are misleading on mixtures of trees”. In: *Science* 309.5744 (2005), pp. 2207–2209.
- [NS08] Arvind Narayanan and Vitaly Shmatikov. “Robust De-anonymization of Large Sparse Datasets”. In: *Proceedings of the 2008 IEEE Symposium on Security and Privacy*. 2008, pp. 111–125.
- [PKK+14] Joon S. Park, Kevin A. Kwiat, Charles A. Kamhoua, Jonathan White, and Sookyoung Kim. “Trusted Online Social Network (OSN) services with optimal data management”. In: *Computers & Security* 42 (2014), pp. 116–136.
- [Pau09] Ohm Paul. “Broken Promises of Privacy: Responding to the Surprising Failure of Anonymization”. English. In: *UCLA Law Review*, Vol. 57, p. 1701, 2010 (2009).
- [SZW+11] Alessandra Sala, Xiaohan Zhao, Christo Wilson, Haitao Zheng, and Ben Y. Zhao. “Sharing Graphs Using Differentially Private Graph Models”. In: *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*. 2011, pp. 81–98.
- [SY13] Entong Shen and Ting Yu. “Mining Frequent Graph Patterns with Differential Privacy”. In: *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2013, pp. 545–553.
- [SMG+12] Maria E. Skarkala, Manolis Maragoudakis, Stefanos Gritzalis, Lilian Mitrou, Hannu Toivonen, and Pirjo Moen. “Privacy Preservation by k-Anonymization of Weighted Social Networks”. In: *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*. 2012, pp. 423–428.
- [Sog08] Chris Soghoian. *Google Log Anonymization*. <http://www.cnet.com/news/debunking-googles-log-anonymization-propaganda/>. 2008.
- [SKX+12] Yi Song, Panagiotis Karras, Qian Xiao, and Stéphane Bressan. “Sensitive Label Privacy Protection on Social Network Data”. In: *SSDBM*. 2012, pp. 562–571.
- [SMJ10] Anna C. Squicciarini, Shehab Mohamed, and Wede Joshua. “Privacy policies for shared content in social network sites”. In: *The VLDB Journal* 19.6 (2010), pp. 777–796.
- [Swe02] Latanya Sweeney. “K-anonymity: A Model for Protecting Privacy”. In: *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* 10.5 (2002), pp. 557–570.
- [TK59] John W. Thibaut and Harold H. Kelley. *The social psychology of groups*. Wiley, New York, 1959.
- [WSL12] Ting Wang, Mudhakar Srivatsa, and Ling Liu. “Fine-grained Access Control of Personal Data”. In: *Proceedings of the 17th ACM Symposium on Access Control Models and Technologies*. 2012, pp. 145–156.
- [WW13] Yue Wang and Xintao Wu. “Preserving Differential Privacy in Degree-correlation based Graph Generation”. In: *TDP* 6.2 (2013).

- [WWW13] Yue Wang, Xintao Wu, and Leting Wu. “Differential Privacy Preserving Spectral Graph Analysis”. In: *PAKDD*. 2013.
- [WB90] Samuel D. Warren and Louis D. Brandeis. “The Right to Privacy”. In: *Harvard Law Review* (1890), pp. 193–220.
- [Wei99] Gerhard Weiss, ed. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT Press, 1999.
- [WYW10] Leting Wu, Xiaowei Ying, and Xintao Wu. “Reconstruction from Randomized Graph via Low Rank Approximation.” In: *SDM*. 28, 2010, pp. 60–71.
- [WYW+11] Leting Wu, Xiaowei Ying, Xintao Wu, and Zhi-Hua Zhou. “Line Orthogonality in Adjacency Eigenspace with Application to Community Partition”. In: *IJCAI*. 2011.
- [WXW+10] Wentao Wu, Yanghua Xiao, Wei Wang, Zhenying He, and Zhihui Wang. “K-symmetry Model for Identity Anonymization in Social Networks”. In: *Proceedings of the 13th International Conference on Extending Database Technology*. 2010, pp. 111–122.
- [XAT12] Qian Xiao, Htoo Htet Aung, and Kian-Lee Tan. “Towards ad-hoc circles in social networking sites”. In: *DBSocial*. 2012, pp. 19–24.
- [XCT14] Qian Xiao, Rui Chen, and Kian-Lee Tan. “Differentially private network data release via structural inference”. In: *SIGKDD*. 2014.
- [XT12] Qian Xiao and Kian-Lee Tan. “Peer-aware collaborative access control in social networks”. In: *CollaborateCom*. 2012, pp. 30–39.
- [XWT11] Qian Xiao, Zhengkui Wang, and Kian-Lee Tan. “LORA: Link Obfuscation by RANdomization in graphs”. In: *Secure Data Management*. 2011, pp. 33–51.
- [YPW+09] Xiaowei Ying, Kai Pan, Xintao Wu, and Ling Guo. “Comparisons of Randomization and K-degree Anonymization Schemes for Privacy Preserving Social Network Publishing”. In: *Proceedings of the 3rd Workshop on Social Network Mining and Analysis*. 2009, 10:1–10:10.
- [YW08] Xiaowei Ying and Xintao Wu. “Randomizing Social Networks: a Spectrum Preserving Approach”. In: *SDM*. 2008, pp. 739–750.
- [YW09] Xiaowei Ying and Xintao Wu. “Graph Generation with Prescribed Feature Constraints”. In: *SDM*. 2009, pp. 966–977.
- [YCY10] Mingxuan Yuan, Lei Chen, and Philip S. Yu. “Personalized Privacy Protection in Social Networks”. In: *Proc. VLDB Endow.* 4.2 (2010), pp. 141–150.
- [ZP08] Bin Zhou and Jian Pei. “Preserving Privacy in Social Networks Against Neighborhood Attacks”. In: *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*. 2008, pp. 506–515.
- [ZPL08] Bin Zhou, Jian Pei, and Wo shun Luk. “A brief survey on anonymization techniques for privacy preserving publishing of social network data”. In: *SIGKDD Explor. Newsl* (2008).

- [ZCO09] Lei Zou, Lei Chen, and M. Tamer Özsu. “K-automorphism: A General Framework for Privacy Preserving Network Publication”. In: *Proc. VLDB Endow.* 2.1 (2009), pp. 946–957.