SECURE UNLINKABILITY SCHEMES FOR PRIVACY PRESERVING DATA PUBLISHING IN WEIGHTED SOCIAL NETWORKS

CHONG KAH MENG

FACULTY OF SCIENCE
UNIVERSITI MALAYA
KUALA LUMPUR

2020

SECURE UNLINKABILITY SCHEMES FOR PRIVACY PRESERVING DATA PUBLISHING IN WEIGHTED SOCIAL NETWORKS

CHONG KAH MENG

DISSERTATION SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE

INSTITUTE OF MATHEMATICAL SCIENCES
FACULTY OF SCIENCE
UNIVERSITI MALAYA
KUALA LUMPUR

2020

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **CHONG KAH MENG**

Registration/Matric No.: **17043420/1 SMA170021**

Name of Degree: **MASTER OF SCIENCE**

Title of Dissertation ("this Work"):

SECURE UNLINKABILITY SCHEMES FOR PRIVACY PRESERVING DATA PUBLISHING IN WEIGHTED SOCIAL NETWORKS

Field of Study:

APPLIED MATHEMATICS

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date: 29/11/2020

Subscribed and solemnly declared before,

Witness's Signature Date: 30/11/2020

Name:
Designation:

SECURE UNLINKABILITY SCHEMES FOR PRIVACY PRESERVING DATA PUBLISHING IN WEIGHTED SOCIAL NETWORKS

ABSTRACT

Preserving privacy of users has been one of the important research issues in social networks. Social networks contain sensitive personal information that are often released for business and research purposes. The privacy of a user can be breached if the data are not released in an anonymized form. In this thesis, we address edge weight disclosure, link disclosure and identity disclosure problems in publishing weighted network data. To counter these privacy risks while preserving high utility of the published data, we define two key privacy properties, namely *edge weight unlinkability* and *node unlinkability*. We design two novel anonymization schemes namely *MinSwap* and δ-*MinSwapX*. The proposed work satisfy the unlinkability notions so that no auxiliary information could be utilized to discover the true edge weight, true link and true identity of a targeted individual with high probability. The edge weight is modified based on minimum value change in order to preserve the usefulness and properties of the edge weight data. In addition, edge randomization is performed to minimally modify the structural information of a user. Experimental results on real data sets show that our schemes efficiently achieve data utility preservation and privacy protection simultaneously.

Keywords: Privacy, utility, weighted social networks, unlinkability, randomization.

SKIM KETANPANAMAAN YANG SELAMAT UNTUK PEMELIHARAAN PRIVASI DALAM RANGKAIAN SOSIAL BERBERAT

ABSTRAK

Pemeliharaan privasi pengguna merupakan suatu isu penyelidikan yang penting dalam rangkaian sosial. Rangkaian sosial mengandungi informasi sensitif yang sering dikongsi untuk tujuan perniagaan dan penyelidikan. Privasi pengguna dapat dibocorkan jika data tersebut tidak diterbitkan dalam keadaan awanama. Dalam tesis ini, kami mempertimbangkan masalah kebocoran berat hubungan, kebocoran hubungan dan kebocoran identiti pengguna semasa penerbitan data rangkaian sosial berberat. Bagi menyelesaikan tiga masalah privasi tersebut dan memelihara utiliti data yang diterbit, kami mentakrifkan dua ciri privasi baru yang mustahak, iaitu ketidak-hubungkaitan berat dan ketidak-hubungkaitan identiti. Dua skim baru dibina berdasarkan ciri-ciri privasi tersebut untuk mengelakkan kebocoran privasi yang berlaku kerana informasi tambahan yang diperolehi oleh musuh. Dalam skim yang dicadangi, berat hubungan dimodifikasikan berdasarkan perubahan nilai yang minima untuk memelihara keaslian dan sifat penting data. Selain itu, hubungan dalam rangkaian asli dimodifikasikan secara rawak and minima untuk memelihara sifat struktur pengguna dalam rangkaian sosial. Keputusan eksperimen yang melibatkan data tulen dan data sintetik berskala menunjukkan bahawa skim yang dibina memelihara utiliti data dan melidungi privasi pengguna.

Kata kunci: Privasi, utiliti, rangkaian sosial berberat, ketidak-hubungkaitan, kerawakan.

ACKNOWLEDGEMENTS

It has been like a journey, I am indebted to many people for their help and support along the way to this dissertation. I would like to express my sincere gratitude to my supervisors, Dr. Amizah Malip and Associate Professor Dr. Wan Ainun Mior Othman, for their priceless supervision, support and patience throughout the work. Despite being extremely occupied with other research and teaching, they always made themselves available for our research meetings and patiently responded to all my inquiries through emails. Their great inspiration, precise guidance and constant encouragement have helped me a lot to improve important skills as a researcher. My tremendous gratitude to my family for their wholehearted encouragement and financial support. They are my great source of motivation and strength. I am deeply indebted to Mr. Wong Yik Chun for the assistance of implementing the algorithms into Python. Last but not least, my special thanks to those who have helped me either directly or indirectly throughout this thesis work.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

D_a	:	Degree of node a
$\overline{D}$	:	Degree sequence
$\boldsymbol{E}$	:	Set of edges
$F_n(a)$	:	Hub fingerprint of node a
G_a	:	1-neighborhood graph of node a
m	:	Number of edges in a network
m_{Add}	:	Number of fake edges added
n	:	Number of nodes in a network
n_{Add}	:	Number of fake nodes added
N	:	Number of distinct edge weight values
$N(Z_p)$	:	Frequency set
$N(Z_T)$	:	Complete frequency set
$\boldsymbol{S}(a,b)$	:	Candidate set
S_a	:	Subgraph of node a
$\boldsymbol{W}$	:	Weight sequence
$\boldsymbol{W'}$	:	Perturbed weight sequence
$\boldsymbol{W(a)}$	:	Set of edge weights associated with node a
$\boldsymbol{W(a \cup b)}$	:	Set of edge weights associated with node a and node b
$W(a,b)$	:	Edge weight from node a to node b
$W'(a,b)$	:	Perturbed edge weight from node a to node b
$w_{i,j}$	:	Edge weight between node i and j

w_p	:	Edge weight in weight sequence for $p = \{1, 2, 3, ..., m\}$
w'_p	:	Perturbed edge weight in weight sequence for $p = \{1, 2, 3, ..., m\}$
$\boldsymbol{V}$	:	Set of nodes
Z_p	:	Possible set
Z_T	:	Universal set
GDPR	:	General Data Protection Regulation
PII	:	Personally Identifiable Information
PPDP	:	Privacy Preserving Data Publishing
K-S	:	Kolmogorov-Smirnov

LIST OF APPENDICES

CHAPTER 1: INTRODUCTION

In this chapter, the motivation of our research is presented. The privacy and utility challenges on privacy preserving social network data publication are discussed. The scope, objectives and contributions of the research are also presented.

1.1 Motivation

A social network is an online platform which enables people to create relationships virtually amongst registered users. With the rapid growth of Web 2.0, social network applications have developed substantially over the last few years (Boulianne, 2019; Newman et al., 2016). Some popular social networks such as Facebook, YouTube, WhatsApp, WeChat, Instagram and TikTok have gained tremendous popularity as these networks support a variety of attractive features and services that help to connect the people (Clement, Feb 14, 2020). The social network users are required to register a virtual profile as their online representation by providing certain basic details such as name, age, address, contact number, email address and other sensitive information. Various types of interaction are performed with other users in the network through sharing of data, ideas and thoughts in the form of videos, photos, text messages, voice messages, posts and files. Huge amount of data generated from the users' activities are digitally collected and shared to third party recipients to enable meaningful data analytics (O'Dea, Feb 28, 2020).

The availability of large scale social network data has driven new business opportunities and research domains. User data such as race, economic status, gender, age, level of education, employment, preferences, interests, browser history, purchase history and other recent activity data are mainly analyzed for efficient marketing and advertisement targeting (Park et al., 2016; Shen et al., 2016). Furthermore, network data are utilized for research purposes in academic communities to extract hidden patterns and behaviours of real

world communities. For example, the connections between users are analyzed to study the formation of communities (Jan & Vlachopoulos, 2019; Kotani & Yokomatsu, 2018), network information spread (Kumbhojkar et al., 2018) and disease control (Arruda et al., 2016; F. S. Lu et al., 2019). Other applications of network data include opinion modeling (Xiong et al., 2017), criminal analysis (Berlusconi et al., 2016; Burcher & Whelan, 2018), shortest paths analysis (Atzmueller et al., 2016; Gong et al., 2016) and spanning trees analysis (Saoud & Moussaoui, 2016; Zhang et al., 2016).

Nevertheless, data sharing may violate the privacy of users as social network data contain sensitive information of the users which should not be disclosed to the public. There are several laws and guidelines enforced by the governments in different countries to restrict the types of publishable data and agreements on the usage and storage of network data. For instance, some significant privacy laws include General Data Protection Regulation (GDPR) in European Union (Tesfay et al., 2018; Voigt & Von dem Bussche, 2017), Act on the Protection of Personal Information in Japan (Fukuta et al., 2017), Information Technology Act in India (Kalia et al., 2017) and Personal Data Protection Act 2010 in Malaysia (Gan et al., 2018). Despite this, a number of real world privacy breaches had occurred due to improper data sharing (Cadwalladr & Graham Harrison, 2018; Causey et al., 2016; Isaak & Hanna, 2018; Novak & Vilceanu, 2019; Trautman & Ormerod, 2016). Such unauthorized disclosure of sensitive data may result in serious implications to both network users and data publishers such as social embarrassment, scam, frauds, stalking, blackmailing, torn reputation, physical threats and economic espionage.

Therefore, it is evident that current privacy laws do not sufficiently protect the data privacy and this motivates us to address the privacy issues associated with network data publication from a scientific perspective.

1.2 Problem Overview

Under the privacy policies compliance, data are released to third party recipients without compromising the privacy of network users. In most instances, data collectors which are simultaneously both service providers and data publishers, may have some specific interests in certain analysis outcome of their data. However, due to lack of skills and in-house expertise to conduct the analysis, outsourcing the task to a third party often become a practical alternative option. In another situation, data are released to the public as required by the organizations and government. For example, patent and trademark information (Q. Lu et al., 2017; Martens, 2018; Wallace & Reinman, 2018) and voter registration data (Berent et al., 2016; Merivaki & Smith, 2020; Pettigrew & Stewart III, 2017). Data are also sold for profit to third parties such as analytics companies, marketing companies and commercial data brokers (Jones & Tonetti, 2019; Lawrenz et al., 2019; Zhu, 2019).

Although data sharing is a common practice to allow data mining, publishing data without sufficient anonymization may prompt to unintended privacy disclosures as sensitive information are not explicitly defined in the network data. Data anonymization is a process of protecting private information by modifying personally identifiable information (PII) to output an anonymized data that cannot be associated with a particular individual. PII is any data that could potentially identify a specific real world individual. Some of these information include name, social security number, date and place of birth, mother's maiden name, biometric records or any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information (McCallister et al., 2010). However, there are some underlying PII that are not explicitly defined in the social network data and could be exploited to compromise the sensitive information of users. For example, the structural information that describes how network users are connected in a network, is one of the PII that is not explicitly defined in the data. Hence,

an efficient anonymization scheme is necessary to release a secure anonymized version of an original database to protect the privacy of a user while serving a variety of data analysis simultaneously.

Privacy of a user and utility of a published data are two contradictory aspects: the privacy is protected by destroying the data, however the data should not be extensively modified to preserve the utility. Bridging both aspects is a challenge since a slight modification on network data could impact a large distortion on data accuracy. Hence, the question on how to minimize the trade-off between privacy and utility is a vital issue in social network data publication. In the next subsections, the privacy and utility challenges of network data publication are discussed.

1.2.1 Privacy Challenges in Network Data Publishing

A privacy breach is defined as a disclosure of personal information that a user intends to keep private from an entity which is not authorized to access or have the information (Tesfay et al., 2018). As more social network data are released publicly, there is a growing concern about privacy breaches for the users involved. The privacy breaches in social networks can be categorized into three types: identity disclosure, link disclosure and edge weight disclosure.

• **Identity Disclosure.** Identity disclosure or node reidentification occurs when the true identity of a targeted individual is revealed by an adversary from a published data. In other words, an individual is reidentified when an adversary is able to map a record in the published data to its corresponding user with high probability. Identity disclosure implies the breach of content information in the profile, which could consist of PII and information about their friends. Furthermore, identity disclosure implies the existence of a user in a particular network. Generally, a user may has strong privacy preference of

their identity in certain sensitive networks such that their existence should not be disclosed. Such networks are common in some social network applications nowadays, where the users could create or join a community group or page that covers a variety of sensitive themes such as politics, sex, religion or healthcare. Reidentification of an individual in such networks would vigorously violate the privacy of users. Identity disclosure is a major privacy issue in social networks as it usually leads to more disclosures such as link disclosure and edge weight disclosure as discussed below.

• **Link Disclosure.** Identity disclosure is associated with the recognition of a user in a network. Meanwhile, link disclosure is another type of privacy disclosure which is associated with the sensitive relationship in a network. Link disclosure is the inference of true link between two users in a published data. In other words, a link is reidentified when an adversary is able to infer the existence of a relationship between two users with high probability. A link is created when a user forms a relationship with other users. Some relationships are safe to disclose to the public such as co-authorship. However, some are sensitive links that may compromise the users upon disclosure such as the relationship of two users in a dating app, financial transaction or homosexual network. Hence, a higher privacy preference may be desirable such that none of their links would be disclosed.

• **Edge Weight Disclosure.** Edge weight disclosure is associated with the intensity of relationship in a network. Edge weight disclosure is the leak of true weightage of an edge to an adversary. In other words, an edge weight is reidentified when an adversary is able to infer the true edge weight value of a user from the published data with high probability. Edge weight implies the intensity, strength, closeness or affinity of relationship between two users, where a user intends to keep it as private information. In most financial networks, edge weights which represent the transaction amounts are usually published.

Therefore, the published data should not only protect the privacy of a node and its link, but also the edge weight.

A published data is said to be **privacy preserved** if an adversary cannot infer the identity, links and edge weight values of a targeted network user from the released data with high probability. The privacy of a user is protected by limiting the ability of an adversary to infer these information, given that the adversary has full access to the published data.

1.2.2 Utility Challenges in Network Data Publishing

Another objective of network data publication is to enable useful data analytics of the original database. In this context, the data accuracy should be preserved post anonymization at an acceptable level according to the intended data utility. Data anonymization involves modification of an original database through different techniques in order to conceal the sensitive information of a user. Hence, the anonymization process may constitute to a certain level of information loss, depending on the extent of modification. More modifications usually lead to higher information loss, which further implies lower data utility but higher data privacy. Social network data consist of structural data which represent the connections of nodes in a network and edge weight data which represent the intensity of connections. This thesis considers three types of data utility which are commonly deployed in previous work to measure the information loss in an anonymized database:

• **Statistical Properties.** Edge weight data reflect the strength of relationships in a network. For example, higher edge weight in a messenger network indicates higher active usage of users in the application. Hence, the global statistical properties of edge weight data should be preserved to allow accurate analysis of user's activities in an application. Some common metrics to quantify the statistical properties are distribution, mean, mode, median,

standard deviation, skewness and kurtosis (Huang et al., 2017; Supriya et al., 2016; Tang et al., 2016).

• **Graph Topological Properties.** Structural data reflect the topological characteristics of a social network (Al-Garadi et al., 2018; K. Das et al., 2018; S. Peng et al., 2018). At macro level, the overall structure of a network is analyzed to study the connection pattern using clustering coefficient. At micro level, the importance and influence of a user in the whole network is measured using centrality such as degree, betweenness, closeness, neighborhood connectivity and edge betweenness. This analysis drives to a more efficient social targeting of advertisements.

• **Shortest Path Analysis.** Shortest path analysis is an important application of network data, which determines the paths in a network that require the least sum of edge weight (S. Liu et al., 2019; Strang et al., 2018; Ventrella et al., 2018). For example, in a shopping network where the edge weight represents the price of items, a buyer always demands on the lowest price from an arbitrary seller. This requires the preservation of shortest path length from the buyer to all the sellers.

Given an arbitrary query to an original database and its anonymized database, the output of query to both databases should be almost similar, that is the difference between the outputs should be less than a user-defined parameter. A **utility preserved** anonymized data could be produced by minimally modifying the edge weight data and network structure so that the published data remain accurate and meaningful in the data mining process.

1.3 Social Network Data Publication

In this section, we present an overview of privacy preserving data publishing (PPDP) environment in a social network. A typical data publishing scenario is centralized, static

and involves three parties: social network users, service provider and data recipients as shown in Figure 1.1. A network user is a data provider who agrees to disclose their data generated in an online social network application to a service provider. On the other hand, a service provider is a trusted entity who operates an online social network application and collects information about their users. The collected data are stored in a form of database, which is then published to third parties for further data analytics. However, the trust relationship is not transitive to third party data recipients. Some data recipients (adversaries) are dishonest and attempt to infer hidden sensitive information of a specific user from the published data. Hence, an original database is required to undergo anonymization prior data publication to permit useful data mining and protect the sensitive information of a user. In the overall data publishing architecture, the privacy of a user is protected by the privacy laws and anonymization processes simultaneously. Further discussions on the components of network data publication are presented in the next subsections.

1.3.1 Social Network Users

A social network user is a real world entity that uses an online network service such as an individual, organization and community. Upon agreement to the terms and conditions issued by the service provider, a user can then enjoy the services provided in the associated online network applications by registering a profile as their virtual representation. The network profile usually contains personal identifiable information, quasi-identifiable information and sensitive information provided by the user. Personal identifiable information is information that uniquely identifies a user, such as social security number, name, email address, mobile number and driving license number. Quasi-identifiable information is information that cannot uniquely identify a user, but potentially identify the user if combined with other auxiliary information such as home address,

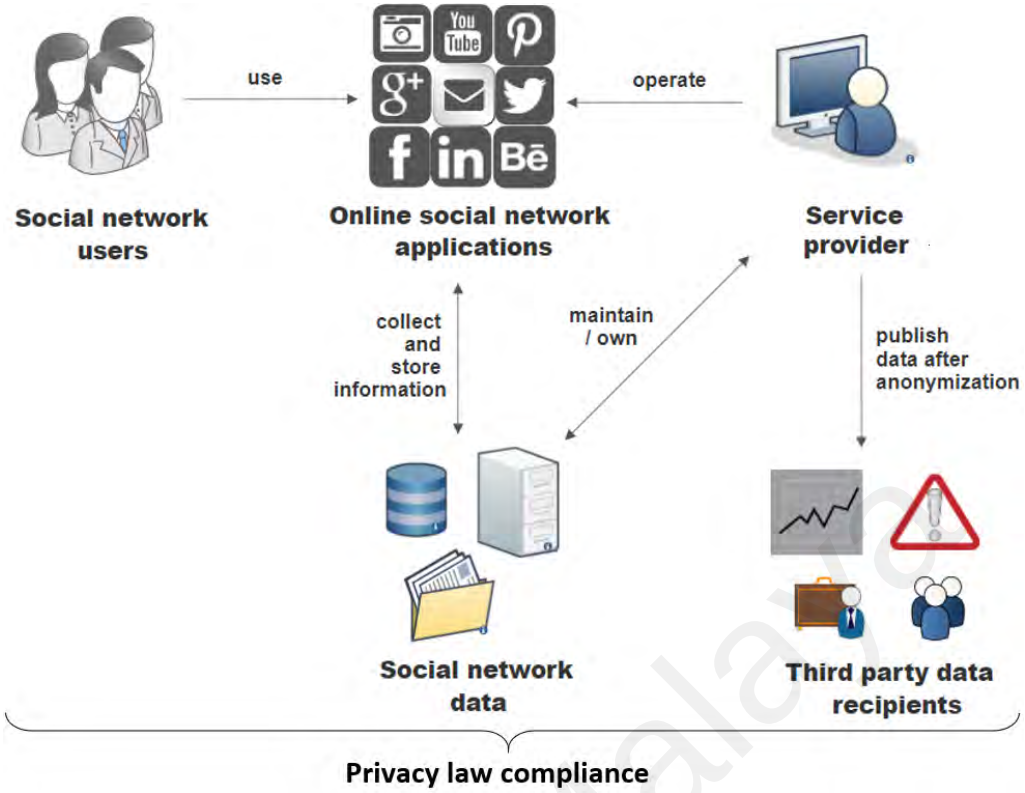

Figure 1.1: Outline of privacy preserving data publishing (PPDP).

postcode, gender, date of birth, former company and hobby. Sensitive information is information that a user intends to keep private from unauthorized parties, which may include religion, political view, salary (as in a financial network) and health conditions (as in a healthcare network). These data could be generated from the users' social activities, which include IP address, location, shopping habit, browsing history and preference.

1.3.2 Service Provider

A service provider is a vendor that builds, develops, operates, manages and delivers a service to a network user. Upon agreement to the terms and conditions of the service provided, a service provider acts as a trusted entity who could legally collect and share selected information of their users. A service provider may collect information about the pages, accounts, hashtags, shares, likes, comments, impressions, URL clicks, keywords and groups that a user connected to and how the user interacts with other users across the services provided. Furthermore, contact information could be collected if a user chooses

to upload, sync or import it from a device. User experience is also collected, especially on how a user uses the services, such as the types of content a user viewed or engaged with, the features a user used the most and the time, frequency and duration of a user's activities. Thus, a service provider is simultaneously a data collector and a data publisher, who compiles data generated by the users and release the aggregated data to the third parties. This is a common social media business model (Bouwman et al., 2018; Di Gangi & Wasko, 2016; Saura et al., 2019; Villi & Picard, 2019), where revenue is mainly generated via advertisements (Facebook, YouTube and Google Ads), e-commerce (Lazada, Shopee, eBay and Taobao) and subscription (LinkedIn Premium and SlideShare Premium). To guarantee the privacy of a user, the data are published post anonymization.

1.3.3 Third Party Data Recipients

A third party data recipient is a natural or legal person, public authority, agency or body other than the network users and service providers, to which the personal data are disclosed. There are two categories of data recipient: honest and dishonest data recipient. An honest data recipient is an entity that does not intend to breach the sensitive data of a user, beyond what is provided in the released data. On the other hand, a dishonest data recipient (adversary) is an entity who attempts to derive sensitive information of a network user from the published data.

1.3.4 Social Network Data

Social network data is defined as a non-directed and weighted graph, $G = (\boldsymbol{V, E, W})$, where the graph consists of edge weight data $\boldsymbol{W}$ and structural data $\boldsymbol{E}$. Structural data describe how the nodes in a network are being connected. The nodes or vertices of a graph, $\boldsymbol{V} = \{v_1, v_2, v_3, \ldots, v_n\}$ denote meaningful entities from the real world such as individuals, organizations and communities. An edge or link $e_{i,j} = (v_i, v_j) \in \boldsymbol{E}$ is an

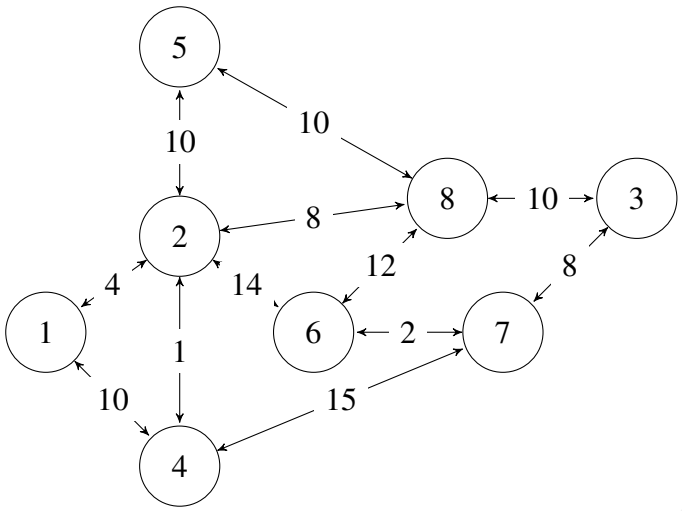

Figure 1.2: An example of a non-directed and weighted social network.

Table 1.1: Representation of Figure 1.2 in tabular form.

No.	Source, v_i	Target, v_j	Edge Weight Value, $W(v_i, v_j)$
1	2	4	1
2	6	7	2
3	1	2	4
4	2	8	8
5	3	7	8
6	5	8	10
7	1	4	10
8	2	5	10
9	3	8	10
10	6	8	12
11	2	6	14
12	4	7	15

association between two nodes $v_i \times v_j \in \boldsymbol{V} \times \boldsymbol{V}$ such as friendship, kinship, partnership, co-authorship, co-workership and transaction between any two entities.

A non-directed graph consists of edges that do not have a direction. The edges indicate a two-way connection between a source node and a target node such that each edge can be traversed in both directions, which imply $e_{i,j} = e_{j,i}$. For instance, a mutual friendship is a non-directed edge. In contrast, a directed graph consists of edges with direction, which indicates a one-way connection from a source node to a target node such that each edge can only be traversed in a specific direction. An example includes a transaction from one

$Node$	1	2	3	4	5	6	7	8
1	0	4	0	10	0	0	0	0
2	4	0	0	1	10	14	0	8
3	0	0	0	0	0	0	8	10
4	10	1	0	0	0	0	15	0
5	0	10	0	0	0	0	0	10
6	0	14	0	0	0	0	2	12
7	0	0	8	15	0	2	0	0
8	0	8	10	0	10	12	0	0

Figure 1.3: Representation of Figure 1.2 in an adjacency matrix.

entity to another. In a weighted network, each edge $e_{i,j}$ is associated with a weight value $w_{i,j} \in \boldsymbol{W}$ which represents the affinity of connection between nodes v_i and v_j, for example, the communication frequency between individuals, degree of friendship, trustworthiness or transaction amount. Figure 1.2 illustrates an example of a non-directed and weighted network. The nodes are indicated by the circles and the edges are represented by the lines.

Network data are represented using either tabular form or adjacency matrix. Figure 1.2 is represented in a tabular form as shown in Table 1.1. An edge between node 2 and 4 (structural data) with value 1 (edge weight data) is written as $W(2, 4) = 1$. Figure 1.3 shows an 8×8 adjacency matrix of Figure 1.2, where the size of the matrix is the number of nodes, the entries represent the edge weight values and entry 0 indicates there is no edge between the two nodes. The nodes can be represented using numbers or alphabets.

1.3.5 Online Social Media Applications

A user interacts with other users in various forms of social network applications. Generally, there are eight types of social network applications according to the functions and characteristics of the applications: social networking sites, professional, academic and informational communities, content communities, blogs, collaborative projects, online shopping networks, consumer review networks and virtual game communities, as shown in Figure 1.4.

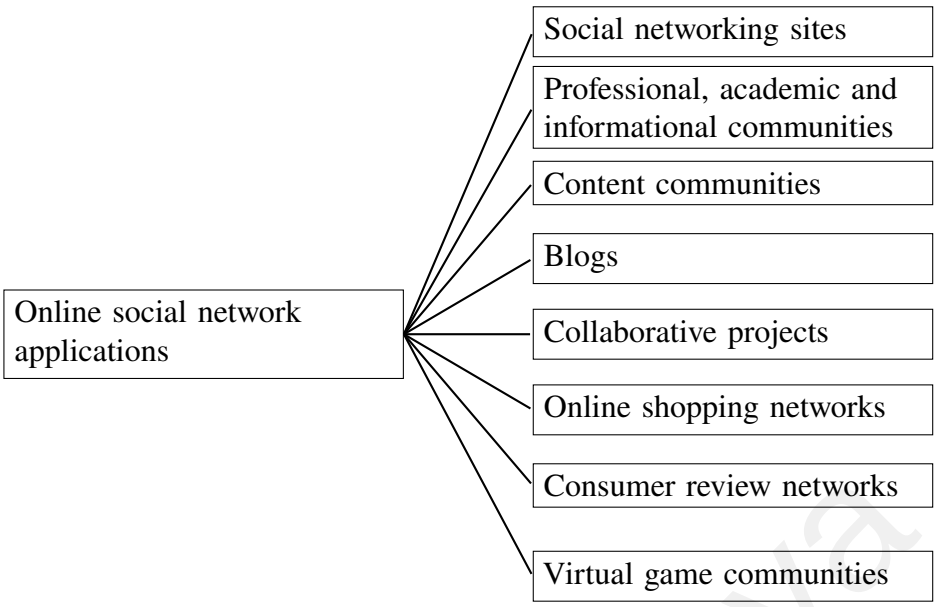

Figure 1.4: Types of social media applications.

Social networking site is the most common platform to share information and idea, where a user interacts with others by inviting friends, sending instant messages or sharing photos, videos, links and files within a particular friend zone (friend list or neighborhood). Some examples of social networking sites are Facebook, Twitter, Sina Weibo and QQ.

Professional communities provide career-related opportunities to a user, where an entity with specific occupations or interests builds connections by joining relevant groups. Examples may include LinkedIn, Opportunity, Sumry, Meetup, Jobstreet and Jobcase. Meanwhile, academic communities are platforms for academic researchers and educators to find, organize, share and review research results to achieve academic advancement. Some examples are Academia.edu, B Academics and ResearchGate. Informational or educational communities such as Codecademy, GeeksforGeeks, Quora, Brilliant, Stack Exchange and Stack Overflow are forums to gain and share knowledge, where users can ask questions and contribute unique insights and useful suggestions.

Content communities allow the sharing of media contents between users. These contents include videos (YouTube, Netflix and Dailymotion), photo (Flickr, Pinterest and Instagram), text (Google Play Books, Scribd and Kindle), PowerPoint presentation (SlideShare, Google

Slides and Prezi), music (Joox, Spotify and SoundCloud) and software (Softonic, Google Play and App store). Social interaction is performed when a user comments, download, upload, rate and share content provided by other users.

A blog is an online website that displays information in chronological order. It is a platform for a writer to express and share their view on a specific issue or it serves as a personal diary that describes the writer's life and experience. Some examples of popular blogs are LiveJournal, Blogger, WordPress and Tumblr. Interaction is performed by leaving a comment on other users' blog.

Collaborative projects allow the joint and simultaneous creation of knowledge-related content by many users to provide a higher quality, compact, self-content and complete content. Some well-known collaborative networks are Encyclopedia, Wikipedia, Google Docs and Wikis.

Online shopping network is another popular form of social network application in which a user (buyer) makes purchases with another user (seller). Interaction is performed when there is a transaction between the users. Examples include Lazada, Zalora, Shopee and Taobao.

In consumer review networks, a user can rate, share or review a brand, product, service and organization. By obtaining and analyzing consumer's review, consumer satisfaction is measured and the future behaviours of the consumer can be modeled and predicted. Instances of this type of networks are Amazon Customer Reviews, TripAdvisor, Zamato and Google Maps Reviews.

In virtual game communities, a user is required to follow the rules of a multiplayer online role-playing game. Interaction is performed when a user chats and completes missions together with their friends in the game. Examples of virtual game communities include World of Warcraft, Grand Theft Auto, PlayerUnknown's Battlegrounds and Fortnite.

1.4 Adversarial Background Knowledge and Attack Model

In this section, we model the potential background knowledge of an adversary and discuss how the background knowledge could be utilized to infer the identity, links and edge weights of a target user in a network.

1.4.1 Adversarial Background Knowledge

In a published data, naive anonymization is applied to remove the explicit identifiers of a user, being replaced with random pseudo-identities. This process protects the data privacy thoroughly when an adversary has zero knowledge of the target individual. However, this is not a realistic assumption as an adversary can exploit different auxiliary information about a target to launch some privacy attacks. This auxiliary information is called background knowledge, which could be collected by investigating the overlapping membership of several social network applications, stealing the browsing history or real-life inspection. In a social network, the background knowledge that could be utilized to intrude user privacy includes edge weight information and structural information. Structural information is the implicit graph information, which is categorized into the following four types:

1. **Degree information**. Degree of node a, D_a is the number of edges connected to the node a. Other variations of degree information are degree pair and degree sequence. A degree pair, (D_a, D_b) is the degree of nodes a and b, while degree sequence, $\overline{D}$ is a monotonic non-increasing sequence of the degree of all nodes in the network.
2. **1-neighborhood graph,** G_a. It is the structural graph up to the first neighborhood of node a.
3. **Subgraph,** S_a. It is a partial network graph that involves node a and some of its edges.
4. **Hub fingerprint,** $F_n(a)$. It provides the information about the distance between a hub a (a node with high degree and high betweenness exceeding the average degree

and betweenness of the network) and other nodes.

We focus on edge weight and structural information as two types of potential background knowledge of an adversary, which are commonly deployed in the current literature (Chen & Zhu, 2015; Cheng et al., 2010; Hay et al., 2007; Li et al., 2017; Q. Liu et al., 2016; Tai et al., 2011; Zhou & Pei, 2011). It is relatively less difficult to collect accurate edge weight information and structural graph of a targeted individual, compared to other types of implicit information (such as eigenvector, betweenness and closeness centrality) (Q. Liu et al., 2016; Zhou & Pei, 2011).

1.4.2 Adversarial Attack Model

In this subsection, we discuss some of the possible attack models to reidentify identity, links and edge weights of a target node as commonly deployed in prior research (Chen & Zhu, 2015; Cheng et al., 2010; Hay et al., 2007; Li et al., 2017; Q. Liu et al., 2016; Tai et al., 2011; Zhou & Pei, 2011).

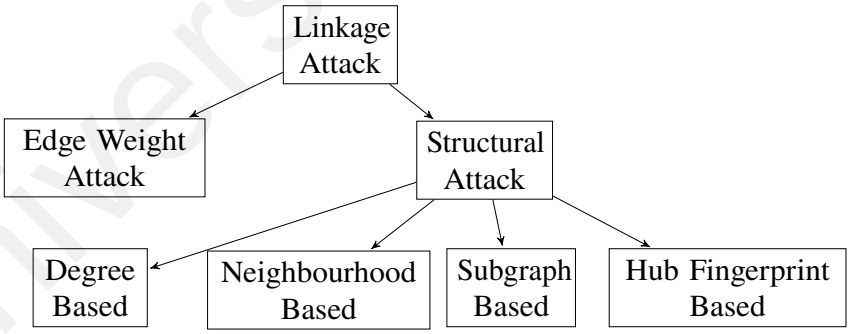

Figure 1.5: Taxonomy of attack models.

Linkage attack is an attack model where an adversary attempts to match the auxiliary background knowledge obtained from external resources to the published data in order to learn some useful information about a target victim. In our work, the published data consists of edge weight and structural data only, other auxiliary information (such as the node label and edge label) provide very little additional information about the nodes in the

published data and thus irrelevant. Figure 1.6 shows a naively anonymized graph of Figure 1.2, where the identities of all nodes are hidden. However, it is insecure when an adversary learns that a node a has two connections of edge weights 1 and 4, then a's true identity (node 2 in Figure 1.2) is revealed. In some cases, edge weight and structural information are combined to reidentify the target. For instance, although node 1 and 5 in Figure 1.2 have similar1-neigbourhood graph (and thus invulnerable to 1-neigbourhood attack), they can be distinguished if an adversary possesses additional background knowledge of edge weight data.

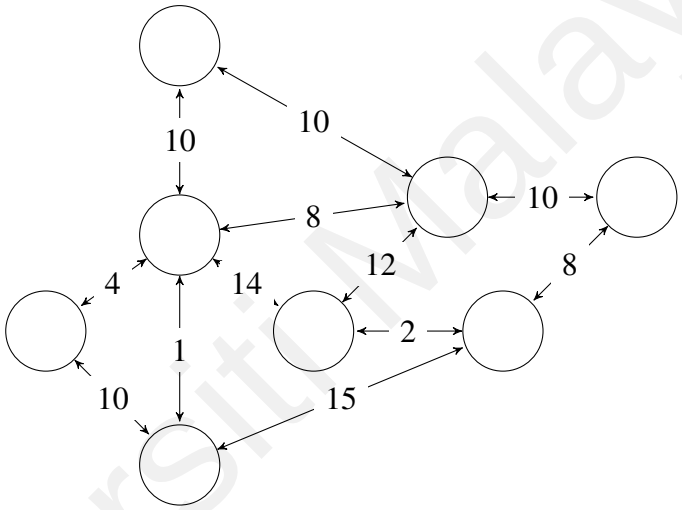

Figure 1.6: A naive anonymized weighted social network of Figure 1.2.

1.5 Scope, Objectives and Contributions of the Thesis

The scope of the thesis focuses on data anonymization schemes on a weighted social network. A secure network data anonymization scheme allows the publication of anonymized data which protects sensitive information of a user and permits useful data analytics on the released data. We concentrate on edge weight disclosure, link disclosure and identity disclosure problems associated with the publication of a weighted network data. We assume an adversary is equipped with complete background knowledge of structural information and edge weight data of a real world target and has full access to the released data. An adversary attempts to infer the identity, links and edge weights of

a target victim in the published data. Based on these realistic assumptions, we propose anonymization schemes that efficiently address the three privacy issues and preserve the data accuracy simultaneously.

Privacy of a user and utility of a published data are two critical concerns in the design of social network anonymization schemes. The objectives of the thesis are as follows:

1. To construct network data anonymization schemes that are secure, usable, efficient and comparable to existing schemes;
2. To enhance data privacy protection against sensitive edge weight disclosure, link disclosure and identity disclosure;
3. To enhance data utility for the preservation of statistical properties, topological properties and shortest path lengths of network data.

Below are the key contributions of the thesis:

1. This thesis defines two new privacy notions that provide additional layers of edge weight unlinkability and node unlinkability to enhance the data privacy, which has not been addressed in prior work.
2. This thesis deploys perturbation and randomization in the design of two new anonymization schemes using the proposed privacy notions that enhance the data utility preservation.
3. This thesis provides a thorough analysis on the anonymization strength of the proposed work and presents extensive experimental results on scalable real data sets that validate the efficiency of our schemes.

1.6 Organization of Thesis

This thesis is organized as follows:

Chapter 2 reviews current literature on privacy preserving social network data anonymization schemes. The chapter is presented in two parts. The first part discusses prior structural anonymization schemes while the second part discusses recent edge weight anonymization schemes according to their modification approaches. We highlight the strengths and limitation of these schemes and present a summary at the end of the chapter.

Chapter 3 presents some relevant mathematical background that are used in this thesis, which include basic set theory, functions, probability theory, descriptive statistical analysis, social network analysis and shortest path analysis.

Chapter 4 presents the proposed work that address three privacy issues in social network data publishing, which include edge weight disclosure, link disclosure and identity disclosure. We designed two novel anonymization schemes, namely ***MinSwap*** and ***δ-MinSwapX*** using two new privacy notions, namely ***edge weight unlinkability*** and ***node unlinkability***. These schemes efficiently achieve high data privacy and high utility preservation simultaneously for a secure and useful data publishing. Simulations using scalable real data sets are performed to validate the efficiency of the proposed work. Furthermore, security analysis is performed to evaluate the privacy protection rendered by our work. The schemes presented in this chapter has been submitted to an IEEE journal as stated below:

- K.M Chong and A. Malip. Trace Me If You Can: An Unlinkability Approach for Privacy Preserving in Weighted Social Networks, *Transactions on Knowledge and Data Engineering*, 2020 (submitted).

Chapter 6 summarizes our key contributions and discuss some future directions of the research.

CHAPTER 2: LITERATURE REVIEW

This chapter presents an in-depth literature review on existing anonymization schemes to guarantee privacy in releasing network data. We focus on a weighted graph data in social networks, which is a combination of structural data and edge weight data. In the first part, we classify prior structural anonymization schemes according to their modification techniques. In a similar manner, we classify recent edge weight anonymization schemes according to their modification approaches. We closely examine the advantages and disadvantages of each scheme and discuss the overall strengths and limitations of the anonymization schemes. Finally, the extent of privacy protection rendered by the existing schemes are summarized in the end of this chapter.

2.1 Structural Anonymization

This section focuses on the existing structural anonymization schemes in a social network. Structural information of a node describes how the node is connected to other nodes in a graph. To protect a user against identity disclosure and link disclosure, the structural information of all nodes should be modified before the data publication.

The structural modification approaches can be grouped under three main classifications: graph modification method, clustering based method and differential privacy method, which are discussed in the next subsections. Figure 2.1 shows the overview of structural anonymization.

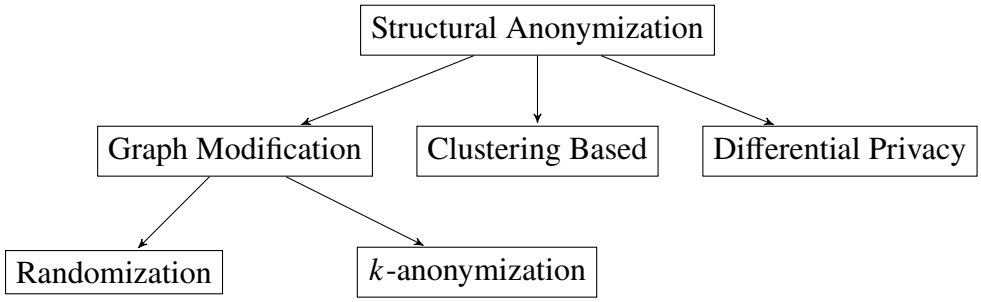

Figure 2.1: Overview of structural anonymization.

2.1.1 Graph Modification

Graph modification approaches anonymize a network by adding, deleting or switching edges or nodes in the original graph. These approaches can be further classified as **randomization**, which performs the graph modification randomly and ***k*-anonymization** method, which performs the graph modification to meet some desired constraints.

2.1.1.1 Randomization

This part discusses the randomization techniques and relevant randomization schemes. The following shows three randomization techniques, which are commonly deployed in most randomization schemes.

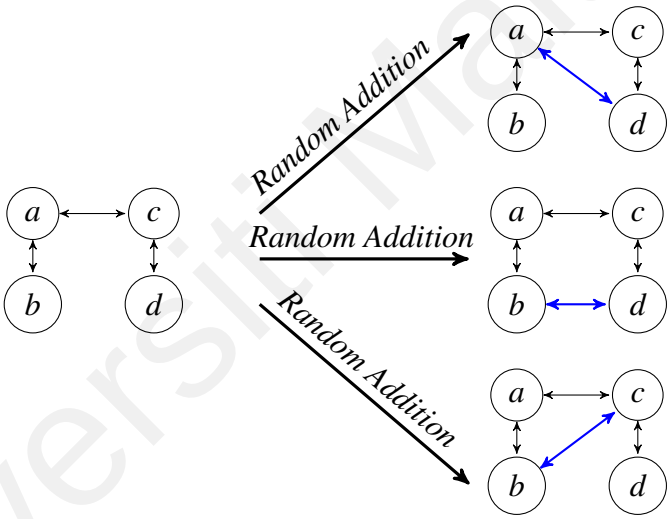

Figure 2.2: Example of a random addition of a fake edge between two existing nodes.

• **Random Addition.** Some non-existing edges are inserted into the original data. These non-existing edges are constructed by forming some fake edges between the existing nodes, as indicated by the blue lines in Figure 2.2. Otherwise, some fake nodes are added into the original graph and edges are formed between the fake nodes and the existing nodes, as shown in Figure 2.3.

• **Random Deletion.** Some existing edges are removed from the original graph, as demonstrated in Figure 2.4.

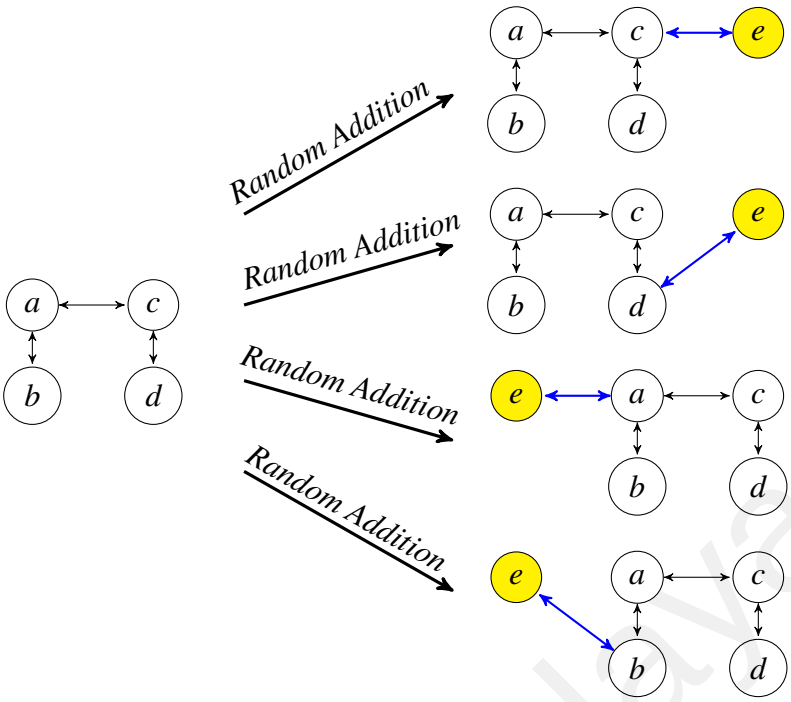

Figure 2.3: Example of a random addition of a fake edge between a fake node and an existing node.

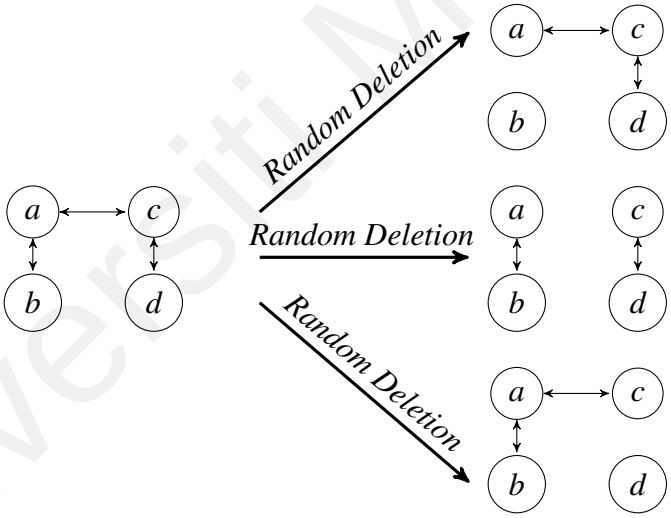

Figure 2.4: Example of a random deletion of an existing edge.

• **Random Swapping.** Two existing edges (a, b) and (c, d) are selected randomly and switched across the node pair to construct two new fake edges (a, c) and (b, d) or (a, d) and (b, c) that do not exist in the original graph, as shown in Figure 2.5. The total number of edges in the original data and degree of each node are preserved.

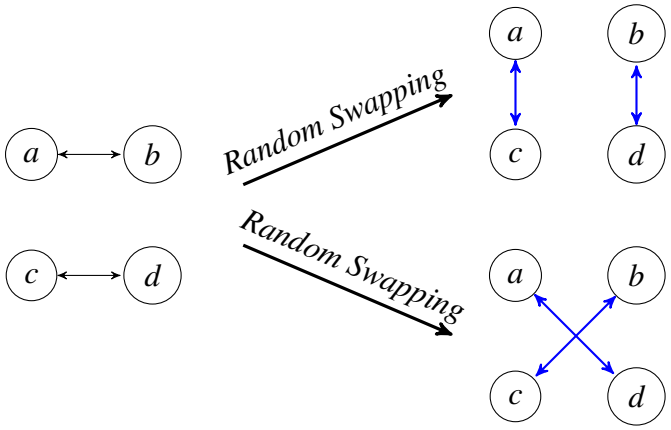

Figure 2.5: Example of a random swapping between two existing edges.

Identity disclosure problem using vertex refinement queries and subgraph knowledge queries as background knowledge was studied by Hay et al. (2007). Vertex refinement queries provide the local structural information of a target node at an increasing power. At level one, the query simply returns the degree of the target node. At level two, the query returns the list of degree of each neighbors of the target node. The subsequent queries could be defined iteratively. Subgraph knowledge queries return the existence of a subgraph around a target node. A k-candidate anonymity was presented, which requires that there are at least k different nodes that match every structural query in the anonymized graph. Hence, the risk of reidentification is capped at $1/k$. The original graph is modified through a series of n random edge deletions, followed by n random edge addition to preserve the number of edges. Although the notion can potentially reduce the risk of identity disclosure, no concrete algorithm was provided to guarantee an anonymized graph that satisfies k-candidate anonymity. Furthermore, it does not guarantee the preservation of data utility.

Two randomization schemes (Ying & Wu, 2008) were proposed to address both identity disclosure and link disclosure problems. *Spctr Add/Del* was presented to preserve the spectral properties of a graph. The spectrum of a graph is the set of eigenvalues[1] of the

[1] If A is an $n \times n$ matrix, then λ is an eigenvalue of A if $AX = \lambda X$ for some non-zero matrix X.

graph's adjacency matrix, which is important to some topological properties of the graph. *Blockwise Random Add/Del* was then developed to modify a set of nodes with high risk of reidentification. This is achieved by clustering all original nodes into blocks according to the degree sequence, followed by a series of edge additions and deletions on the set of selected nodes. Although a greater extent of utility could be preserved, some unmodified nodes are not protected through the anonymization process, which contradicts the laws and privacy policies. The work of Ying and Wu (2008) was then enhanced for implementation in a weighted social network (Q. Liu et al., 2016).

Another randomization scheme (P. Liu et al., 2017) was proposed to address identity disclosure and link disclosure problems. Bernoulli distribution is deployed to modify the edges instead of random edge addition and deletion. Bernoulli distribution is the probability distribution of a single experiment that produces outcome "1" or "0" only with probability p and $q = 1 - p$, respectively. The original network data is modeled as an adjacency matrix A, which contains entry $a_{ij} = 1$ if there is an edge between node i and j, or entry $a_{ij} = 0$ if there is no edge between node i and j. Different p (probability of retaining an existing edge) and r (probability of adding a non-existing edge) of Bernoulli trials are simulated to each entry of matrix A to output a new matrix A'. The entries of matrix A' determine the removal or insertion of each edge in the original data. Nevertheless, the parameter setting of p and r for each entry of matrix A poses a challenge. It is cost inefficient as the number of parameters defined is high in a scalable network. In addition, optimal pair of p and r to optimize anonymized graph in terms of privacy and utility is difficult to compute.

Fard and Wang (2015) focused on the link privacy protection and presented a *neighborhood randomization* scheme to preserve the global graph structure. Given an original edge (i, j), conventional randomization probabilistically randomizes this edge to a new fake (i, k) without considering the structural proximity of nodes, where node k may be structurally

far from node i. Hence, the original structure of a network could be significantly distorted and the anonymized data yields less accurate social network analysis. The proposed solution randomizes an edge by restricting the randomization to the neighboring nodes of i. An edge (i, j) is retained with a probability p and a fake edge (i, k) is formed with a probability $1 - p$. The fake node k is randomly selected from a local D-neighborhood that is structurally close to node i (a set of nodes that are D length from node i and $D > 1$). Thus, a sensitive edge is hidden and the network structure could be preserved to a greater extent.

Discussion on Randomization Schemes

Randomization is a standard technique to achieve identity and link privacy. It provides some attractive features compared to other methods. Firstly, randomization is simple, flexible and feasible to be implemented practically. Furthermore, it does not focus on the adversary's background knowledge as the sensitive information of a user in the randomized graph are protected through the random process that modifies the graph. An adversary could not confidently infer the identity and links of a user with high probability as the association rules between background knowledge and sensitive information are dimmed. Succinctly, randomization provides a meaningful level of privacy protection to a user, regardless of the amount of auxiliary structural information possessed by an adversary.

However, the existing randomization schemes do not guarantee data utility. For example, the edge deletion is unpredictable. Some important nodes and edges could be removed from the graph, which drastically distorts the network topology and the shortest paths. The edge addition is also erratic. Suppose a high number of fake edges are added to the same node, then the local properties of the node are significantly altered. Hence, the randomization should be adapted in line with some utility-preserving constraints to retain a desired level of data usefulness.

2.1.1.2 *k*-anonymization

A k-anonymity (Sweeney, 2002) is a widely used privacy model in anonymizing relational data (such as healthcare data in tabular form). It guarantees that there are at least k indistinguishable records in the published data through suppression (replace the values with asterisk '*') and generalization of the attribute values (modify numerical values into an interval that contains the exact values). Hence, any individual cannot be reidentified from the published data with a probability of higher than $1/k$.

This part discusses the variations of k-anonymity model in social networks. In a k-anonymization method, the edges and nodes in the network are modified to produce multiple indistinguishable nodes and edges to achieve certain privacy requirements. Different assumption on an adversary's background knowledge leads to different expectation of privacy criteria. The schemes are grouped according to the types of structural information, which can be classified into the following four categories: degree of nodes, D-neighborhood graph, subgraph and hub fingerprint.

1. **Degree of nodes**. It is the number of edges incident to a node a, denoted by $D_a = |\{b|(a, b) \in \boldsymbol{E}\}|$ of a graph $G = (\boldsymbol{V}, \boldsymbol{E})$. For example, $D_a = 3$ and $D_b = 2$ in Figure 2.6. Degree pair of a and b, (D_a, D_b) is the degree information of node a and b, which is (3, 2). Degree sequence, $\overline{D}$ is a monotonic non-increasing sequence of the degree of all nodes in a network. $\overline{D} = \{2, 2, 3, 3, 4\}$.
2. **Subgraph**. It is a partial network graph that consists of a particular node and some of its edges. A subgraph of node a is denoted by $S_a = \{(V_{S_a}, E_{S_a})|a \in V_{S_a} \subset \boldsymbol{V} \wedge (a, b) \in E_{S_a} \subset \boldsymbol{E}\}$, where V_{S_a} are the nodes of subgraph S_a and E_{S_a} are the links of subgraph S_a. Figure 2.7 shows some weighted subgraphs of node a in Figure 2.6.

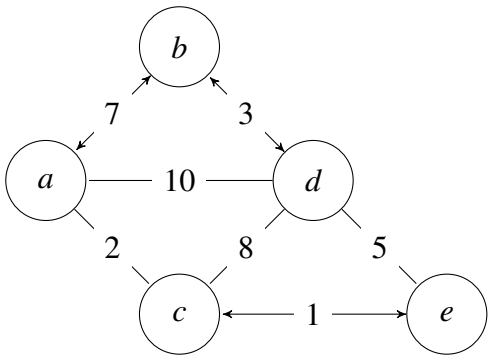

Figure 2.6: Another example of weighted network.

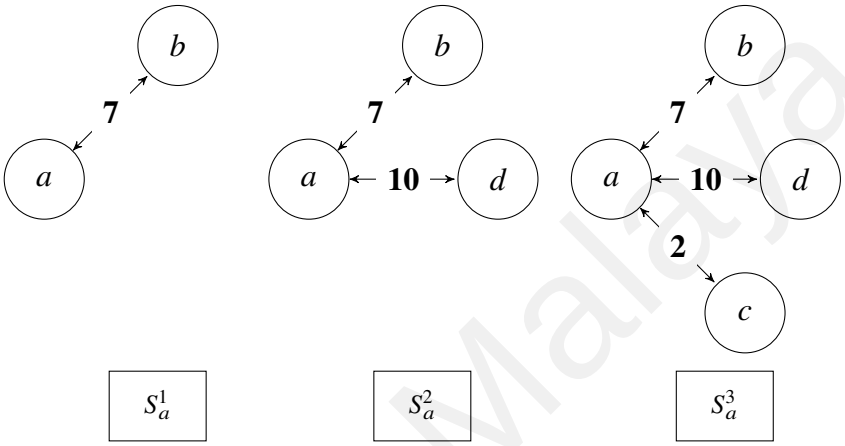

Figure 2.7: Subgraphs of node *a*.

3. ***D*-neighborhood graph**. A D-neighborhood of node a is the set of all nodes that lie within D-hops from node a. It is a subset of subgraph. One node has only one neighborhood graph. For example, S_a^3 in Figure 2.7 is the 1-neighborhood graph of node a. When $D > 1$, D-neighborhood graph is difficult to be collected and the accuracy of the collected graph is questionable in a scalable network. Hence, an adversary has to obtain information about many nodes to initiate D-neighborhood attacks, for $D > 1$, which is often infeasible in real world scenarios. A 1-neighborhood graph is utilized instead to attack the published data.
4. **Hub fingerprint**. A hub is a node with high degree and high betweenness centrality in a network. A hub fingerprint describes the shortest path length between a set of designated hubs and other nodes in a network. The hub fingerprint of node a is denoted by $F_n(a)$, where n is a limit on the maximum hop of observable hub

connections. For instance, consider node a and d in Figure 2.6 as two hubs, hub fingerprint of node e is a vector of the shortest path lengths (bounded by n) to each hub. $F_1(e) = (0, 5)$ because node e is not connected to node a in one hop or less but it is 5 distance away from node d. $F_2(e) = (3, 5)$ because node e is 5 distance away from node d and is 3 distance away from node a in two hops or less (shortest path of node a and e is $e \rightarrow c \rightarrow a$).

• **Degree Based Anonymization.** A graph-anonymity model called k-degree anonymity was proposed to tackle the identity disclosure problem with a background knowledge of node's degree information only (K. Liu & Terzi, 2008). The k-degree anonymity requires that there are at least k nodes with the same degree in the published graph. The proposed model consists of a two-step anonymization algorithm. During the first step (degree sequence's anonymization), a new degree sequence that satisfies k-degree anonymous is constructed by solving a linear-time dynamic programming. Then, a minimal number of edges are added into the graph to achieve k-degree anonymous. The algorithm is extended by modifying the graph through edge deletions, edge swapping and simultaneous edge additions and deletions to achieve k-degree anonymous. This work was further improved in terms of time complexity (Bhattacharya & Mani, 2015) and feasibility in anonymizing large-scaled data (Casas Roma et al., 2013). Bhattacharya and Mani (2015) proposed an iterative algorithm with linear time complexity to construct a k-anonymous degree sequence, compared to the quadratic time complexity of prior k-degree anonymity. Another enhanced degree sequence's anonymization called Univariate Micro-aggregation for Graph Anonymization (UMGA) was proposed to anonymize scalable networks with less number of edge modifications (Casas Roma et al., 2013).

Another type of degree based attack called friendship attack was studied (Tai et al., 2011), where an adversary possesses the degree of two connected nodes and attempts to

reidentify the corresponding victims from the published data. To protect against such attack, a k^2-degree anonymity was proposed, which requires that for every node with an incident edge of degree pair (D_a, D_b) in the published network, there exist at least k-1 other nodes with the same degree pair. An integer programming and heuristic approach were deployed in the construction of a k^2-degree anonymous graph. However, determining an optimal solution has an exponential time complexity and is computationally infeasible for scalable data.

Degree of nodes provides limited structural information of a target victim. Degree attack could be launched and rectified easily by either randomization or k-anonymization. Although the degree based schemes above are invulnerable to degree attack, they are insecure against other stronger structural attack models. This motivates the design of stronger anonymization schemes, discussed in the following parts.

• **Neighbourhood Based Anonymization.** A neighborhood attack is a node reidentification attack using knowledge about the directly connected neighbors of a targeted node and the relationship among the neighborhood. This attack was first addressed by proposing a k-neighborhood anonymity model (Zhou & Pei, 2008), which guarantees that there exist at least k indistinguishable nodes in the published graph such that the 1-neighborhood graphs of each k nodes are all similar. The algorithm proposed consists of two phases, which are node grouping and edge modifications. In the first phase, nodes are grouped according to the similarity of their neighborhood graph, which is evaluated using depth-first search tree. Then, each group of nodes undergoes node and edge additions until there are at least k nodes with similar neighborhood in each group. The number of fake node and fake edge is the utility loss metric. A different variation of k-neighborhood anonymity (Bensimessaoud et al., 2016) was designed to enhance the preservation of average path length and protect the data against neighborhood attack.

Another scheme was proposed (Zhou & Pei, 2011) by combining k-neighborhood anonymity and ℓ-diversity[2] (LeFevre et al., 2005; Machanavajjhala et al., 2006), with an additional background knowledge of node's label. The scheme requires that a published graph satisfies k-neighborhood anonymity and contains at least ℓ different node labels. Although it renders stronger privacy level to a user, ℓ-diversity has higher time complexity of $O(n^2/k)$, which compromises its feasibility in real world implementation.

The neighborhood graph of a node is modified such that there are k similar neighborhood graphs in the published data. However, subgraph attack is still possible to reidentify a user as the subgraphs of a user are retained in the published data. Such vulnerability is addressed in the following schemes.

• **Complete Structural Based Anonymization.** Complete structural based anonymization addresses any structural attack, which includes attack models using node's degree, 1-neighborhood graph, subgraph and hub fingerprint. A k-automorphism was proposed (Zou et al., 2009) to defend against identity disclosure problem, which guarantees that there are at least k indistinguishable nodes in the network in terms of structural information. Two graphs are automorphic if and only if the graphs are symmetrical to each other, as shown in Figure 2.8. Hence, any individual cannot be reidentified with a confidence level of higher than $1/k$ using structural information as the background knowledge. The proposed algorithm works with a series of node and edge addition and deletion to construct a k-automorphic graph. The utility loss is reflected by the change of number of edges. Although k-automorphism protects the identity of a user against any structural attack, the network properties are drastically distorted, rendering the data to be useless.

A k-isomorphism was proposed (Cheng et al., 2010) as an enhanced version of k-automorphism to address additional link disclosure problem in social networks. Two

[2] ℓ-diversity requires every attribute value group to contain at least ℓ distinct attribute values.

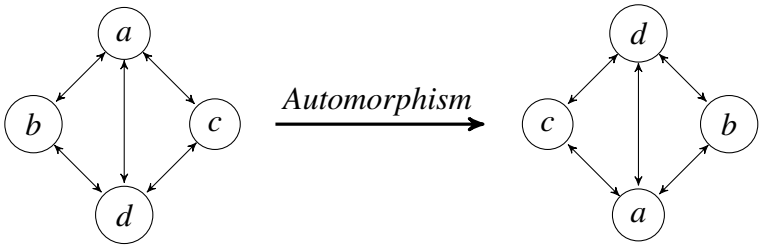

Figure 2.8: Example of a graph automorphism.

graphs are said to be isomorphic if and only if the graphs contain the same number of nodes and the nodes are connected in the same pattern, as shown in Figure 2.9. A k-isomorphism modifies the original graph into k isomorphic subgraphs through several rounds of edge addition. The nodes in the original graph are preserved as no true node is removed from the network. However, the limitation of such modifications is substantial. The determination of automorphic and isomorphic subgraphs is an expensive process, which demerits the practicability of both schemes in scalable networks. Furthermore, the utility loss is large as the graph structure is modified rigorously.

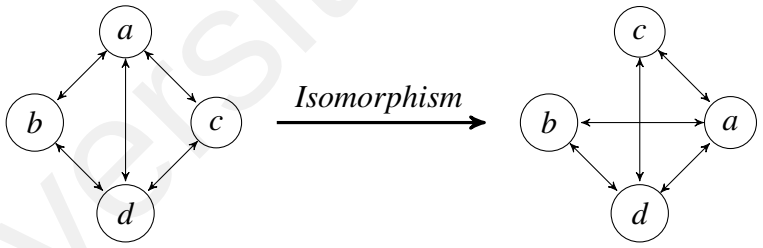

Figure 2.9: Example of a graph isomorphism.

Discussion on *k*-anonymization Schemes

All k-anonymization schemes discussed provide privacy guarantee such that there are at least k matched nodes given a structural query in the published data, which limits the risk of identity disclosure to a maximum level of $1/k$. However, k-anonymization schemes incur unnecessary information loss when the privacy parameter k is high. More edge additions and deletions are performed to achieve k indistinguishable nodes. This would significantly affect the network properties as well as the data usefulness. On the other

hand, if the privacy parameter is low, the schemes would provide insufficient privacy protection to a user. This is called the privacy-utility trade off. In addition, modification of k indistinguishable nodes with respect to the structural graph is practically infeasible due to the high cost and high computational complexity of finding an optimal solution to the problem especially when the network is scalable. Hence, it is not a cost-effective method for network anonymization.

2.1.2 Clustering Based Method

This section discusses the clustering based techniques and relevant clustering schemes. Clustering based method anonymizes a social network by grouping nodes and edges into groups that are called supernodes and superedges, subject to some constraints on the characteristics of the nodes and edges. Clustering techniques were deployed to address identity disclosure (Babu et al., 2013; Campan & Truta, 2008; Hay et al., 2008). The nodes are grouped into some disjoint partitions based on the similarity of their labels and structural information. Then, the number of nodes in each partition along with the density of the edges that exist within and across the partitions are published as an anonymized data. Figure 2.10 illustrates a clustering process involving five nodes. The anonymized graph is a generalized version of the original graph.

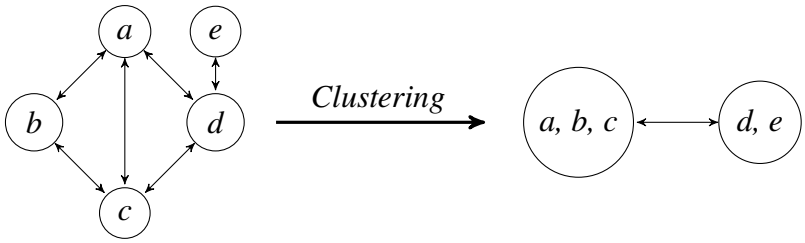

Figure 2.10: Demonstration of clustering based method.

Discussion on Clustering Based Schemes

Merging nodes and edges can effectively prevent identity disclosure and link disclosure as the original nodes and edges cannot be reidentified from the anonymized graph. However, the graph is shrunk post-anonymization and most of the local structures are difficult to be analyzed. Furthermore, the network centrality is not preserved and shortest path analysis would yield inaccurate output, resulting in a low data utility.

2.1.3 Differential Privacy

Differential privacy (Dwork, 2008) provides a formal privacy guarantee to the nodes of a database, despite the auxiliary information available to an adversary. Intuitively, a differential privacy model injects some random noises derived from Laplace distribution or normal distribution to the results of a query performed on a database. Differential privacy guarantees that an adversary in possession of the released results is not able to determine the existence of an individual in the original database. Therefore, the released results provide meaningful interpretations about the underlying population statistics of the database but obscure the presence of any individual. The notion of differential privacy was adapted to network data and two new privacy definitions were formalized, namely edge differential privacy and node differential privacy.

In edge differential privacy (Hay et al., 2009), two graphs G and G' are said to be edge neighbors if G' can be obtained from G by deleting or adding k arbitrary edges from G. Hence, edge differential privacy guarantees that an adversary is not able to infer the existence of a particular edge in an original database G with high probability. Nissim et al. (2007) deployed edge differential privacy to estimate the cost of minimum spanning tree and the number of triangles of a graph and presented an algorithm that returns the smooth sensitivity of statistics. Furthermore, edge differential privacy was improved in terms of the accuracy of answering network queries such as number of triangles and cycles and

general subgroup counts (Rastogi et al., 2009). A local differential privacy model was proposed to preserve community structure information of a centralized and decentralized social graph with higher accuracy (P. Liu et al., 2019; Zhan et al., 2017).

In node differential privacy (Hay et al., 2009), two graphs G and G' are said to be node neighbors if G' can be obtained from G by deleting or adding a single node including all its adjacent edges from G. Hence, node differential privacy guarantees that an adversary is not able to infer the existence of a target node in an original database G with high probability. However, research on node differential privacy mainly focused on improving the accuracy of publishing the degree distribution of a graph (Day et al., 2016; Hay et al., 2009; Macwan & Patel, 2018).

Discussion on Differential Privacy Schemes

In terms of privacy, differential privacy is a strong model as it does not depend on the background knowledge of an adversary. However, the main drawbacks of differential privacy model are presented on the utility aspect. Randomization and k-anonymization methods release a privacy preserved graph which can be studied in place of the original database, to allow a broader range of analysis. Nevertheless, the released results under a particular differential privacy model only can serve a specific query. Furthermore, differential privacy is highly inaccurate to queries with high sensitivity. The sensitivity of a query is the largest possible difference that one data point can affect on the result of that query, for any data set. Some of the high sensitivity queries include clustering coefficient, path length distribution, betweenness distribution and closeness distribution. Removing a node or an edge from a graph may have a catastrophic effect on path lengths in the network, causing some finite lengths to become infinite, and thus drastically alter the clustering coefficient, betweeness and closeness scores. Indeed, publishing individual centrality scores could be very sensitive under both node and edge differential privacy. Moreover,

it is impossible to release a named list of influential individuals under node differential privacy as the existence of an individual should not be inferred in a differentially private database. In general, the current research trend mainly focuses on improving the utility preservation of differential privacy in network graphs.

2.1.4 Overall Discussion of Structural Anonymization Schemes

We present the strengths and drawbacks of current structural anonymization schemes in this discussion. The data privacy is mainly guaranteed based on the anonymity notion in the structural anonymization schemes above. Anonymity means that a node is not identifiable by an adversary in the published data. Meanwhile, data utility is not guaranteed as important nodes and edges could be removed from the original data. Our work fills the gap by proposing a new and secure randomization technique that incurs a lower utility loss. This is achieved by considering edge deletion based on the importance of edges in the original network, such that essential edges could be preserved in the published data. Hence, this may preserve network centrality to a greater extent.

2.2 Edge Weight Anonymization

Although research on structural anonymization are in the mature stage, these work could not be directly applied in a weighted network, which contains an additional edge weight value on every link. Structural anonymization alone does not guarantee sufficient protection to a network user in the presence of edge weight as an additional background knowledge, even though the nodes are structural indistinguishable. Hence, edge weight anonymization should be applied in line with structural anonymization to ensure a secure data sharing.

This section focuses on edge weight anonymization schemes in a weighted social network that are relevant to our work. Data perturbation is one of the most effective

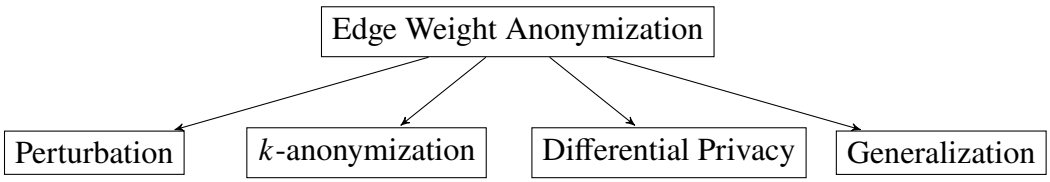

Figure 2.11: Edge weight anonymization models.

methods to modify edge weight values by adding noise to databases to prevent node reidentification (Wilson & Rosen, 2003) while at the same time, maintain the shortest paths characteristic between pairs of nodes in the network. A k-anonymization method modifies edge weight data to achieve k similar edge weights and paths in the published data. Differential privacy is a rigorous privacy model which gives a formal guarantee that edge weight data of a user in the database is not leaked (Chen & Zhu, 2015; Li et al., 2017). Generalization technique is applied in a few schemes only as it guarantees a low utility preservation (Babu et al., 2013; Q. Liu et al., 2016). As the existing edge weight anonymization schemes discussed in this section inherit similar privacy-utility trade off, we shall present the overall discussion towards the end of this review section.

2.2.1 Data Perturbation

Data perturbation is a statistical disclosure control method that modifies a database by adding noises into the database to achieve data privacy. In a social network, data perturbation was utilized in several schemes to insert noises into the edge weights to address edge weight disclosure and identity disclosure problems and preserve shortest paths of the original network.

Identity disclosure problem in a weighted network was studied in L. Liu et al. (2008, 2009). Two privacy strategies were developed for different natures of network. The first one is a Gaussian Random Multiplication Perturbation (GRMP) developed for dynamic networks, which adds Gaussian noise[3] to the original edge weights to achieve shortest path

[3] Gaussian noise is a statistical value derived from normal distribution and is added to modify the original data.

preservation. The distribution of edge weights is considered as a normal distribution based on the mathematical assumption that a data set approaches normal distribution when the size of data increases. Nevertheless, the edge weight is shown to be power-law distributed in most real life scenarios (McGlohon et al., 2011). Hence, the introduction of Gaussian noise does not guarantee the desired privacy and utility preservation of network data if the edge weights are not normally distributed. The second strategy is a greedy perturbation algorithm developed for static networks. This method finds the shortest path length between selected nodes, then adjusts every weight lies on that path (either increases or decreases by a constant amount) so that the path length remains unchanged. However, a target node could be reidentified with high probability by linking the edge weight information to the associated node. Furthermore, both methods have a low performance on the shortest path preservation as they only preserve shortest paths between a small number of selected nodes.

To improve the deficiency of L. Liu et al. (2009), GRMP was enhanced in terms of scalability and shortest path protection by M. Liu et al. (2017). Instead of deriving noise from Gaussian distribution, the noise is derived from centrality features of the network itself such as node degree, betweenness centrality, pageRank[4] and clustering coefficient. Experimental results showed that their work are more effective in the shortest path preservation than GRMP.

A linear programming model was proposed (S. Das et al., 2010a, 2010b) to anonymize the edge weights while preserving the properties of the graph that are expressible as a linear function of the edge weight. The edge weight data are modeled as a matrix and the anonymization is formulated as a linear optimization problem. However, a feasible solution to the optimization problem is not guaranteed especially when the data is scalable,

[4] PageRank is a score used to determine the order in which search engine results are presented.

which compromises the practicability of this method in large networks. In Figure 2.12, a_{ij} is the coefficient obtained from the linear function, x_i is the edge weight value and b_i is the constraints of the linear programming. No feasible solution can be found if the equations are inconsistent. In other words, the equations are contradictory. For example, $-x - y \leq -2 \Rightarrow x + y \geq 2$ and $x + y \leq 1$.

$$\underbrace{\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{km} \end{pmatrix}}_{\mathbf{A}} \underbrace{\begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_m \end{pmatrix}}_{\mathbf{x}} \leq \underbrace{\begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}}_{\mathbf{b}}$$

$$\begin{pmatrix} -1 & -1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \leq \begin{pmatrix} -2 \\ 1 \end{pmatrix}$$

Figure 2.12: An example of linear programming.

2.2.2 *k*-anonymization

A *k*-anonymous weighted edge was proposed (L. Liu et al., 2010), such that there exist at least *k*-1 other edges in which the differences between the edge weights are less than a predefined parameter. A probabilistic graph is used to modify the edge weights so that a limited number of selected shortest paths are preserved. Nevertheless, it is not possible to construct an anonymized graph that is *k*-anonymous with all the shortest paths remain unchanged. The change of edge weight values could be sufficiently large. The initial shortest path in the original network becomes non-shortest path in the published network due to the large increment on the original path length.

A *k*-anonymous path privacy model was presented to protect the sensitive shortest path between two nodes in a weighted graph (Wang et al., 2011). It prevents the true shortest path from being revealed by ensuring that there exists at least *k* shortest paths with

same shortest path distance. Thus, this limits the sensitive path disclosure to a maximum probability of 1/k. The algorithm determines the shortest and the second shortest path in the network. The edge weights of the non-overlapping edges between the shortest and the second shortest paths are then reduced proportionally until the second shortest path distance equals the shortest path distance. This is called as weight-proportional based modification. An enhanced k-anonymous path model was proposed (Wang, Tsai, et al., 2013) to modify the edge weights by considering network centrality such as PageRank and nodes' degree. As it is assumed that edge weights can only be modified once, a k-anonymous path privacy cannot be guaranteed when multiple node pairs are involved. Moreover, the edge weights are modified using a multivariable linear function, which can be utilized to reveal or estimate the original edge weight from the published data. Thus, the proposed methods do not render sufficient node protection to the users.

The k-anonymous path scheme was further improved (Wang, Shih, et al., 2013) with additional background knowledge of nodes' degree on the shortest path. An adversary who possesses the knowledge of nodes' degree on the true shortest path may be able to reveal the true shortest path although the graph satisfies k-anonymous path privacy. Thus, a (k_1, k_2)-shortest path privacy was proposed to ensure that there are at least k_1 indistinguishable shortest paths between the source and target nodes. In addition, for the non-overlapping nodes on the k_1 shortest paths, there exist at least k_2 nodes with same node's degree and lie on more than one shortest path. There are more restrictions on the modification of edge weight, which leads to a greater information loss compared to the k-anonymous path.

A k-anonymous weighted edge and a k-degree anonymity were combined (Yuan & Chen, 2011) to propose a k-weighted-degree anonymous model. The edge weights and nodes' degree were assumed as adversarial background knowledge. This model ensures that in an anonymized graph, there are at least k indistinguishable nodes having the same degree

and the distances between the weight sequence of those nodes are within a predefined constant. After obtaining a new degree sequence that is *k*-degree anonymous using the proposed algorithm in (K. Liu & Terzi, 2008), new edge weight values are assigned to the new created edges. The edge weights are adjusted using a linear programming model based on three distance functions which are absolute distance, relative distance and rate distance to guarantee that the edge weights generated are nearly valued to other edge weights associated to the node.

A scheme was proposed to address identity disclosure in a weighted network by deploying clustering and *k*-anonymization (Skarkala et al., 2012). First, *k*-anonymity is used to group the original nodes into supernodes and original edges into superedges. In the second step, an average edge weight is calculated for each superedge and is then reassigned back to all the edges in the corresponding superedge. However, some unique edge weight values are destroyed and the number of distinct values in the original data is reduced to *k*.

2.2.3 Differential Privacy

Differential privacy (Dwork, 2008) is deployed to modify edge weight data by adding Laplace noise. It guarantees that the statistical properties of a database is insensitive on a record change. Thus, the output probability of the same results will not change significantly, regardless of the presence of a record in the data set. This is a stronger anonymization technique as it makes no assumption about the background knowledge of any potential adversary.

Differential privacy was adopted to preserve the privacy of social recommendation in (Chen & Zhu, 2015). It first clusters the nodes into supergroups, then Laplace noise is added to the average edge weight of each supergroup to modify all edge weights. Another differential privacy scheme was presented to protect the edge weights of social networks and preserve shortest path (Li et al., 2017). The scheme assumed edge weight sequence

as an unattributed histogram. Barrels with the same count are merged into one group to reduce the amount of injected noise. Then, Laplace noise is added to edge weight to guarantee k-indistinguishability between groups so that the number of groups with the same amount of barrels is at least k.

Although differential privacy is powerful theoretically, it has several privacy and utility limitations. Firstly, the original data could be estimated with high accuracy from repeated queries (Tang et al., 2017). If an adversary performs a series of repeated differential privacy queries (k times) on a published database, then the original data could be disambiguated with high probability. Hence, Laplace noises must be injected k times into the original data to guarantee that the published data is invulnerable against k times of such queries. When k is large, the utility of the published data is degraded significantly. In a differentially private database, a maximum of q times queries are allowed to ask the database. This parameter q is called the privacy budget. The privacy of a database cannot be guaranteed if more than q times queries are made to the database. Thus, the database would stop answering further queries and provide no data utility after q times queries.

Differential privacy preserves utility for low-sensitivity queries such as counting, as the presence or absence of a single record changes the result slightly by one. However, a differentially private database could provide extremely inaccurate results for high-sensitivity queries. Examples of high-sensitivity queries include sum, maximum, minimum, averages and correlation. Consider a user with high edge weight value, the removal of such edge weight data from a network database may significantly change the statistical properties of the database. Hence, a differentially private database is expected to provide highly biased results for more complex queries, such as variance, skewness and kurtosis.

2.2.4 Generalization

Generalization replaces an original edge weight value with a generalized value. A generalization approach was deployed in Babu et al. (2013), where the edge weights are recalculated as the ratio per total edge weight. The edges and nodes are clustered into superedges and supernodes. Then, for all edge weights in a superedge, the new edge weight is computed as a fraction of original edge weight over the total edge weight in the superedge. Another generalization scheme was proposed to generalize original edge weights in a superedge into a range of value (Q. Liu et al., 2016). For example, if edge weights 3, 4, 8 and 10 are categorized into a group, then range of value [3,10] is reassigned to the four edge weights. The larger the range, the higher information loss.

Generalization is a simple approach deployed to reduce the probability of node reidentification as there are more nodes possessing the same generalized value in the published data. In addition, the actual edge weight values could not be inferred with high probability as the values are generalized. However, unlinkability is not achieved as the generalized values infer certain relationship between original data and published data. In other words, a user could be linkable to their sensitive information in the published data. Moreover, the new edge weights provide very little specific information regarding the nodes in the original network, which render the published data to be useless.

2.2.5 Overall Discussion of Edge Weight Anonymization Schemes

We summarize the overall strengths and limitations of current edge weight anonymization schemes in this discussion. While anonymity has been addressed in the schemes presented, the aspect of unlinkability has not been considered. Unlinkability is a key property of social network that prevents an adversary from linking the actual edge weight values to its associated nodes. The schemes discussed do not consider the weight linkability property of the network data as the published edge weights could be reverse-engineered to disclose

the original data. The association rule between the original value and the published value is retained in the released data. Hence, the published data certainly leak some useful information about the original data and the noise injected could be estimated, provided the association rule is defined to an adversary. Our work fills the gap of the literature by addressing unlinkability in a social network, which requires that no auxiliary edge weight data could be utilized to infer the original edge weight data and the identity of a user.

In the literature, original data are removed from the database and noise are commonly added to the database to preserve selected shortest path. The performance of these schemes are low as they preserved only a limited number of selected shortest paths. Furthermore, they do not preserve the statistical properties of original data such as mean, median, variance, skewness and distribution. Our work fills the gap by deploying data swapping technique to further preserve the statistics of a database.

2.3 Summary

This chapter closely review related anonymization schemes in a social network. The anonymization techniques discussed in the previous subsections are summarized in Tables 2.1 and 2.2. Privacy preserving social network data publishing remains a challenging problem as it is difficult to propose a feasible model that meets all privacy requirements and utility objectives. It will be interesting to explore new and effective solutions that achieve both privacy and data utility simultaneously in social networks and this is where our work steps in.

Table 2.1: Structural anonymization models.

Privacy Model	Adversary's Background Knowledge				Method			Privacy Preservation	
	D	**S**	**N**	**H**	**GM**	**CL**	**DP**	**ID**	**LD**
k-candidate anonymity (Hay et al., 2007)	✔	✔	✔		✔			✔	
Rand Add/Del (Ying & Wu, 2008)	✔				✔			✔	
Randomized perturbation (P. Liu et al., 2017)	✔				✔			✔	✔
Partial k-anonymity (L. Peng et al., 2017)		✔			✔			✔	
k-degree anonymity (K. Liu & Terzi, 2008)	✔				✔			✔	
k^2-degree anonymity (Tai et al., 2011)	✔				✔			✔	
k-neighborhood anonymity (Zhou & Pei, 2008)			✔		✔			✔	
k-neigbourhood anonymity and ℓ-diversity (Zhou & Pei, 2011)			✔		✔			✔	
k-automorphism (Zou et al., 2009)	✔	✔	✔	✔	✔			✔	
k-isomorphism (Cheng et al., 2010)	✔	✔	✔	✔	✔			✔	✔
(Hay et al., 2008)	✔	✔	✔	✔		✔		✔	✔
SaNGreeA (Campan & Truta, 2008)	✔	✔	✔	✔		✔		✔	✔
Edge differential privacy (Hay et al., 2009) (Nissim et al., 2007) (P. Liu et al., 2019) (Rastogi et al., 2009) (Zhan et al., 2017)	✔	✔	✔	✔			✔		✔
Node differential privacy (Day et al., 2016) (Hay et al., 2009) (Macwan & Patel, 2018)	✔	✔	✔	✔			✔	✔	✔

Notes: **D** denotes degree knowledge, **S** denotes subgraph, **N** denotes 1-neighborhood, **H** denotes hub fingerprint, **GM** denotes graph modification approach, **CL** denotes clustering based approach, **DP** denotes differential privacy approach, **ID** denotes identity disclosure, **LD** denotes link disclosure, ✔ denotes selected.

Table 2.2: Edge weight anonymization models.

Privacy Model	Method	Privacy Preservation			Utility Preservation
		ID	LD	EWD	
Gaussian random multiplication perturbation and shortest path greedy perturbation (L. Liu et al., 2009)	Data perturbation	✔		✔	Shortest path
Anonimos (Das et al., 2010)	Data perturbation			✔	Linear properties of graph
k-anonymous weighted edge (L. Liu et al., 2010)	k-anonymization			✔	Shortest path and shortest path length
k-anonymous path (Wang et al., 2011) (Wang, Tsai, et al., 2013)	k-anonymization		✔	✔	Shortest path and shortest path length
(k_1,k_2)-shortest path privacy (Wang, Shih, et al., 2013)	k-anonymization		✔	✔	Shortest path and shortest path length
k-weighted-degree anonymous model (Yuan & Chen, 2011)	k-anonymization	✔		✔	Minimal weight change
k-anonymization (Skarkala et al., 2012)	k-anonymization	✔	✔	✔	Shortest path
Private recommendation generator (Chen & Zhu, 2015)	Differential privacy	✔	✔	✔	Accuracy of recommendation
Merging barrels and consistency inference (Li et al., 2017)	Differential privacy	✔	✔	✔	Shortest path
Fractional generalization (Babu et al., 2013)	Generalization	✔	✔	✔	Minimal weight change
Probabilistic indistinguishability (Q. Liu et al., 2016)	Generalization	✔	✔	✔	Minimal weight change

Notes: **ID** denotes identity disclosure, **LD** denotes link disclosure, **EWD** denotes edge weight disclosure and ✔ denotes selected. All references in Table 2.2 considered edge weight information as adversary's background knowledge.

CHAPTER 3: MATHEMATICAL TOOLS

In this chapter, we present some basic mathematical backgrounds required in the design of our work, which include set theory, functions and probability theory. Furthermore, we discuss several analysis tools to quantify the information loss induced by the proposed schemes, such as descriptive statistics, network analysis and shortest path analysis. Readers with foundation in calculus, statistics and network graph analysis may skip this chapter as it only functions as an introductory chapter for readers who are not from mathematics background.

3.1 Set Theory and Functions

Set and sequence are used to model edge weight data. In this section, we discuss some basic notation of a set and its operations in the build-up of our schemes. We also present the concept of functions to model the relationship between an original data and an anonymized data in this thesis.

3.1.1 Set Theory

A **set** is a collection of distinct objects, which are called elements. The elements of a set could be any type of object, such as numerical values, non-numerical values, mathematical operations or even other sets. The order of the elements is not important and no repetition of element is allowed. For example, $G = \{1, 2, 3, 4, 5\}$ denotes a set with five elements which are 1, 2, 3, 4 and 5. Set G can also be written as $\{5, 4, 2, 1, 3\}$ as the order of the elements is not important. Set $H = \{1, 1, 1\}$ is not a valid set as the element 1 occurs more than once.

A **sequence** is another type of collection of objects, in which the order of the elements is important and repetitions of element are allowed. The position of an element in a sequence is called **index**. The edge weight data is modeled as a sequence of edge weight values. For

example, $W = \{3, 3\}$ is a valid sequence, where edge weight value at index 1 equals edge weight value at index 2, such that $w_1 = w_2 = 3$. In this thesis, we indicate a set or sequence using capital letters and its elements using small letters.

We have some special sets of number:

1. Natural numbers, $\mathbb{N} = \{1, 2, 3, 4, 5, ...\}$.
2. Integers, $\mathbb{Z} = \{..., -2, -1, 0, 1, 2, ...\}$.
3. Rational numbers are numbers which can be expressed as a ratio between two integers, $\mathbb{Q} = \{\frac{p}{q} \mid p, q \in \mathbb{Z}\}$.
4. Irrational numbers $\mathbb{I}$ are numbers which cannot be expressed as a ratio of two integers.
5. Real numbers are all rational and irrational numbers, $\mathbb{R} = \mathbb{Q} \cup \mathbb{I}$.

We say that $x \in A$ if x is an element of set A and $x \notin A$ if x is not an element of set A. For example, if $A = \{1, 2, 3\}$, then $1 \in A$ and $4 \notin A$. We write $\forall x \in A$ to denote "for all elements x from set A" and $\forall x \notin A$ to denote "for all elements x that are not from set A". Furthermore, we write $\exists x \in A$ to denote "there exists at least one element x from set A" and $\exists x \notin A$ to denote "there exists at least one element x that is not from set A".

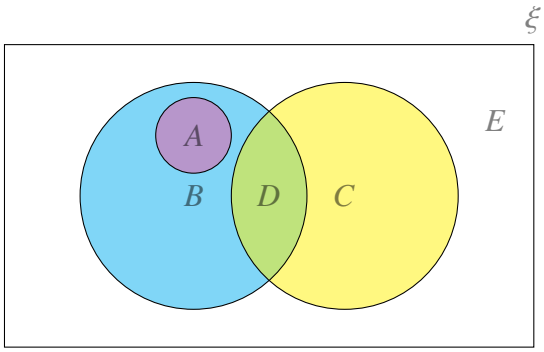

Figure 3.1: Example of a venn diagram.

Let *A, B, C, D* and *E* denote some sets of a venn diagram in Figure 3.1, we say that A is a subset of B if each element of A is also an element of B. Using formal notation, $A \subseteq B$ if $\forall x \in A \implies x \in B$. This is logically equivalent to its contrapositive statement:

$\forall x \notin B \Rightarrow x \notin A$. Set A is a proper subset of set B (denoted by $A \subset B$) if $A \subseteq B$ and $A \neq B$. This implies every element of A is an element of B and there exists at least one element in B which is not an element of A. Using formal notation, $A \subset B$ if $\exists x \in B \wedge x \notin A$.

An empty set is a set containing no element and is denoted by $\emptyset$. An empty set is a subset of every set. A singleton set containing single element x is denoted by $\{x\}$. Clearly $x \in \{x\}$ and $x \neq \{x\}$. Furthermore, $y \in \{x\}$ if and only if $x = y$. Two sets are equal if and only if they contain identical elements, which implies A is a subset of B and B is a subset of A. In formal notation, $A = B \Leftrightarrow A \subseteq B \wedge B \subseteq A$. The universal set, ξ is the set containing all possible elements.

The intersection of two sets A and B is the collection of all elements that exist in both sets. In formal notation, $A \cap B = \{x \mid x \in A \wedge x \in B\}$. The union of two sets A and B is the collection of all elements that exist in either set. It is written as $A \cup B = \{x \mid x \in A \vee x \in B\}$. The set difference, $A - B$ yields the elements in A that are not in B. It is defined by $A - B = \{x | x \in A \wedge x \notin B\}$. In Figure 3.1, $A \cap B = A$, $A \cup B = B$ and $\xi - (B \cup C) = E$.

The complement of a set A, denoted by A^c contains all elements in the universal set that are not in A. In formal notation, $A^c = \{x | x \notin A\}$. The cardinality of a set A is the number of elements in set A, which is denoted as $|A|$. For example, $|\{1, 2, 3\}| = 3$.

Some properties of complement set are:

1. $(A \cup B)^c = A^c \cap B^c$ (DeMorgan's Law)
2. $(A \cap B)^c = A^c \cup B^c$ (DeMorgan's Law)
3. $(A^c)^c = A$
4. $A \subseteq B \Rightarrow B^c \subseteq A^c$ (Contrapositive)

3.1.2 Functions

The relationship between an original data and an anonymized data is modeled as a function (mapping) f. Given w is an edge weight value in an original data W and w' is an

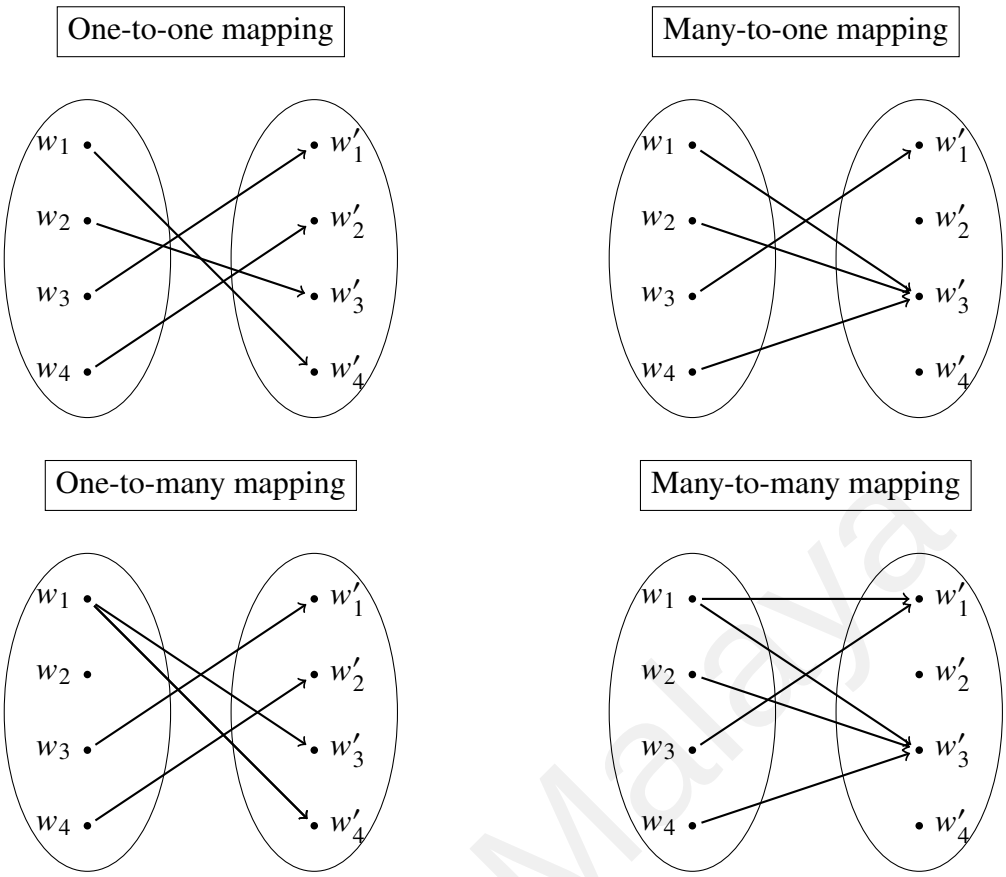

Figure 3.2: Types of mapping.

edge weight value in an anonymized data W', w is mapped to w' under a function f, such that $f : W \rightarrow W'$, for all $w \in W$, $w' \in W'$. We say that w is an object, w' is an image of w under f, W is the domain of f and W' is the codomain of f.

Generally, the mapping could be one-to-one, many-to-one, one-to-many and many-to-many, as shown in Figure 3.2.

- **One-to-one mapping:** Under a one-to-one or injective mapping, there is exactly one corresponding image in the codomain for each object of the domain.
- **One-to-many mapping:** Under a one-to-many mapping, there are at least one image in the codomain for an object of the domain.
- **Many-to-one mapping:** Under a many-to-one mapping, there are at least one object for a given image in the codomain.

- **Many-to-many mapping:** Under a many-to-many mapping, there are at least one image in the codomain for an object of the domain and there are at least one object for a given image in the codomain.

In formal notation, a function f is said to be **injective** if $f(w_1) = f(w_2)$ implies $w_1 = w_2$ for all $w \in W$. Every injective function has its unique inverse function, where each w' is mapped back uniquely to w under a function f^{-1}, such that $f^{-1} : W' \rightarrow W$. Injective mapping is insecure compared to other types of mapping as w' could be reverse-engineered to determine its original unique w. To verify a non-injective mapping, we may prove that there exists at least one image for a given object. In formal notation, $\exists w \in W$ such that $f(w_1) \neq f(w_2)$ when $w_1 = w_2$,

3.2 Probability Theory

The privacy level of our schemes are measured by the probability of privacy breaches. In this section, we discuss the fundamental of calculating the probability of privacy leaks. Some basic definitions are given as follows:

- **Experiment:** An experiment is the process of an operation that leads to results.
- **Outcome:** An outcome is a possible result of an experiment.
- **Event:** An event is a collection of outcomes.
- **Exhaustive event:** A set of events is exhaustive if at least one of the events must occur. For example, the event of tossing a coin are exhaustive as either the outcomes head or tail must occur.
- **Mutually exclusive event:** Two events are mutually exclusive if they cannot both occur (be valid) at the same time. For instance, tossing a coin is a mutually exclusive event.

- **Probability of event A, $P(A)$:** $P(A)$ is the number of ways event A can occur divided by the total number of possible outcomes. $P(A) = 0$ indicates event A is impossible and $P(A)$ indicates event A is certain.
- **Intersection probability, $P(A \cap B)$:** $P(A \cap B)$ is the probability of both A and B occurs.
- **Union probability, $P(A \cup B)$:** $P(A \cup B)$ is the probability of A or B occurs.
- **Complement of probability, $P(A^c)$:** $P(A^c)$ is the probability of A does not occur.
- **Conditional probability, $P(A|B)$:** $P(A|B)$ is the probability of A occurs, given that B occurs. It is defined by:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \tag{3.1}$$

- **Law of total probability:** The law of total probability is given by:

$$P(A) = \sum_n P(A \cap B_n) \tag{3.2}$$

$$= \sum_n P(A|B_n) \cdot P(B_n) \tag{3.3}$$

$$= P(A|B) \cdot P(B) + P(A|B^c) \cdot P(B^c) \tag{3.4}$$

Consider the venn diagram in Figure 3.1, the events A and C are mutually exclusive events. Hence, $P(A \cap C) = 0$ and $P(A \cup C) = P(A) + P(C)$. Meanwhile, the events A and B are not mutually exclusive events. Thus, $P(A \cap B) = P(A)$ and $P(A \cup B) = P(A) + P(B) - P(A \cap B) = P(B)$. The sum of probability of any event A and its complement is one, that is $P(A) + P(A^c) = 1$.

3.3 Descriptive Statistics

Descriptive statistics are used to interpret, describe and understand the properties of edge weight data by drawing short summaries about the sample and measures of the dataset. The preservation of statistical properties of edge weight data allows the inferences of significant conclusions about the hidden patterns underlying the dataset. Generally, there are two main types of descriptive measures used to analyze edge weight data, which include measures of central tendency and measures of spread. We also present a Kolmogorov-Smirnov test (Massey Jr, 1951) to verify the data distribution preservation prior and after anonymization.

3.3.1 Measures of Central Tendency

A measure of central tendency is a single value that describes the central position of a dataset. The central position of a frequency distribution is measured using mean, median and mode.

• **Mean**. The mean is the sum of all edge weight values in a dataset divided by the number of edge weight values in that dataset. Given that we have m edge weight data in a dataset with values $w_1, w_2, w_3, ..., w_m$:

$$\text{Population mean, } \mu = \frac{\sum_{i=1}^{m} w_i}{m} \tag{3.5}$$

• **Median**. The median is the middle edge weight value of a dataset that has been arranged in ascending order of magnitude. For example, for a dataset with values {1, 2, 3, 4, 5}, the median = 3. For a dataset with values {1, 2, 3, 4, 5, 6}, the median = $\frac{3+4}{2}$ = 3.5.

• **Mode**. The mode is the most frequent edge weight value in a dataset. There could be no mode for dataset with all values appear with same frequency, such as dataset {1, 2, 3, 4}. If two values appeared with same maximum frequency, then the dataset has two modes.

For example, dataset {1, 1, 2, 2} has two modes of 1 and 2.

3.3.2 Measures of Spread

A measure of spread describes the degree of spread out of a dataset. We measure this using standard deviation, variance, skewness and kurtosis.

• **Standard Deviation**. Standard deviation is the average distance between each edge weight from the mean. That is, how data are spread out from the mean. A low standard deviation indicates that the edge weights are closer to the mean of the dataset compared to that of high standard deviation. The population standard deviation is given by:

$$\text{Population standard deviation, } \sigma = \sqrt{\frac{\sum_{i=1}^{m}(w_i - \mu)^2}{m}}. \tag{3.6}$$

• **Variance**. Variance is a square of average distance between each edge weight from the mean. Hence, it is the square of standard deviation and is denoted by σ^2.

• **Skewness**. Skewness is a measure of symmetry of a dataset. A dataset is symmetric if mode, median and mean are the same. A normal distributed dataset is symmetric. However, a distribution could be non-symmetric. A distribution is positively skewed if the tail is on the right side of the distribution, that is, mode < median < mean. A distribution is negatively skewed if the tail is on the left side of the distribution, that is, mode > median > mean. Figure 3.3 shows the types of data skewness.

There are two methods to calculate the skewness coefficient of a dataset (Doane & Seward, 2011; Doric et al., 2009):

$$\text{Pearson first coefficient of skewness (mode), } S_{coef} = \frac{\text{Mean - Mode}}{\text{Standard deviation}} \tag{3.7}$$

$$\text{Pearson second coefficient of skewness (median), } S_{coef} = \frac{3(\text{Mean - Median})}{\text{Standard deviation}} \tag{3.8}$$

The direction of skewness is given by the sign. A zero means the data is symmetric. A negative value means the distribution is negatively skewed. A positive value means the distribution is positively skewed. Pearson coefficient compares the data distribution with a normal distribution. Hence, the larger the value, the larger the difference of the data distribution from a normal distribution.

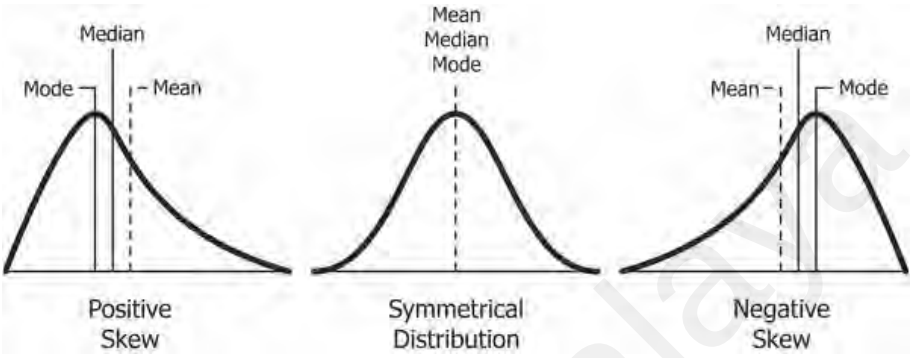

Figure 3.3: Skewness of a data distribution.

• **Kurtosis**. Kurtosis (Dokov et al., 2017) is a measure to evaluate the heaviness of the tail in a frequency distribution. It can be calculated as follows:

$$Kurt[w] = \frac{\sum_{i=1}^{m}(w_i - \mu)^4}{m\sigma^4} \tag{3.9}$$

A positive value indicates a heavy tail, where there are a lot of data in the tail. Conversely, a negative value indicates a light tail, where there are little data in the tail. This heaviness in the tail implies the peakedness of the data compared to a normal distribution.

3.3.3 Two Sample Kolmogorov-Smirnov Goodness-of-Fit Test

The two sample Kolmogorov-Smirnov test (*K-S* test) (Finner et al., 2018) is a common non-parametric goodness-of-fit test that determines whether two datasets come from the same distribution. An advantage of this test is that it can be adopted regardless of the data distribution and sample size. We do not utilize the Chi-squared goodness-of-fit test

(D'Agostino, 2017) as it depends on an assumption of independent normally distributed data. The test is not suitable for work in this thesis as we consider real experimental data with unknown edge weight distribution.

Given that a network dataset with sample size m is equally separated by N ordered edge weight values $w_1, w_2, ..., w_N$ in ascending order, the cumulative distribution function (CDF) is defined as:

$$CDF(i) = \frac{m(i)}{m}, i = \{1, 2, 3, ..., N\} \tag{3.10}$$

where $m(i)$ is the number of edge weight less than w_i.

The first step of *K-S* test is to state a null (H_0) and alternative hypothesis (H_a) for the *K-S* test as shown in Table 3.1. Next, a test statistic, *D-Stat* is defined by evaluating the maximum distance between the two CDFs of the datasets. A test statistic lower than the critical value of selected confidence level, α indicates that both datasets likely follow the same distribution. A critical value is a point on the sample distribution that is compared to the test statistic to determine whether to reject the null hypothesis. The critical value of different α can be referred from Appendix A. The Kolmogorov-Smirnov test is summarized in Table 3.1:

3.4 Social Network Analysis

Social network analysis (SNA) is a tool used to study the topological properties of social networks through the application of graph theory. A node's centrality is a measure of the prominence or structural importance of that node in a network in terms of power, communication, influence, control or status. Determination of the most central nodes in a network helps to improve the effectiveness of information dissemination in a network, advertising targeting, epidemics control and suspected terrorists identification. Unless

Table 3.1: Procedure of *K-S* test.

H_0	Both datasets come from a population with the same distribution.
H_a	Both datasets do not come from a population with the same distribution.
Test Statistic, $D\text{-}Stat$	The Kolmogorov-Smirnov test statistic is defined as the maximum distance between the two CDFs at point i. $$D\text{-}Stat = \max_{1 \le i \le N} \lvert E(i)_1 - E(i)_2 \rvert \quad (3.11)$$
Significance Level	Alpha, α.
Critical Value Test Statistic, $D\text{-}Crit$	Refer to Appendix A, Kolmogorov–Smirnov Table for the corresponding critical value.
Interpretation	It is significant to reject H_0 if $D\text{-}Stat > D\text{-}Crit$. Otherwise, it is not significant to reject H_0 if $D\text{-}Stat \le D\text{-}Crit$.

otherwise stated, the formulas in this section are adopted from (Bloch et al., 2019; K. Das et al., 2018; Rodrigues, 2019). We refer the reader to (K. Das et al., 2018) for a more comprehensive understanding on network centrality as this section only discusses the relevant backgrounds to provide a basis for network centrality used in the thesis.

3.4.1 Network Centrality

We discuss some important network centrality metrics that are commonly deployed to evaluate a network dataset, as presented below:

• **Degree**. Degree centrality of a node a denoted by D_a is the number of edges that are directly connected to node a. Suppose n is the number of nodes in a network, degree centrality could be normalized to obtain a number between 0 and 1 as follows:

$$\text{Normalized degree centrality, } D'_a = \frac{\text{Degree centrality}}{\text{Maximum possible number of edges}} \quad (3.12)$$

$$= \frac{D_a}{n-1} \quad (3.13)$$

Degree centrality provides some insight into the connectivity of nodes in a network. A high degree centrality indicates that the node is popular and resourceful in the network. For example, node a is the most central node in Figure 3.4.

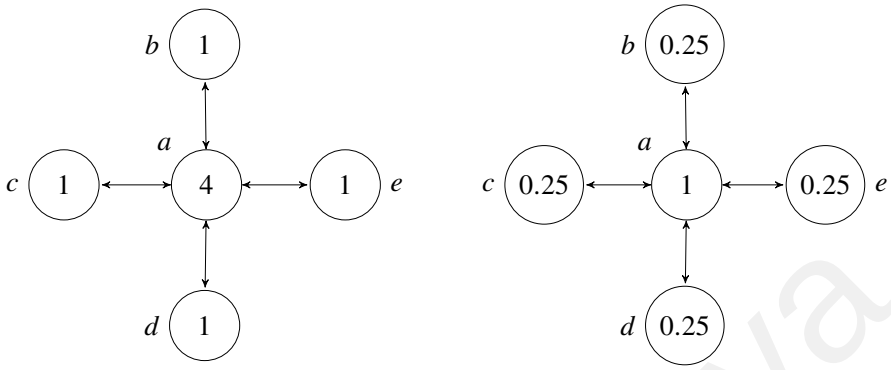

Figure 3.4: Example of degree centrality and normalized degree centrality of a network.

• **Betweenness**. Betweenness centrality measures how well a node is connected to other parts of the network. A node with high betweenness centrality functions as a bridge, broker or gatekeeper in a network as the node is closer to other nodes in the network. Identifying nodes with high betweenness helps to control, disrupt or improve information flow around a network. Betweenness centrality of a node a is calculated by identifying shortest paths (geodesics) of all node pairs and then counting the number of shortest paths that involve node a. Let $\sigma(i, j|a)$ denotes the number of geodesics between node i and j that involve node a, $\sigma(i, j)$ denotes the number of geodesics between node i and j and n_{Max} denotes the maximum possible number of geodesics excluding node a, the betweenness and normalized betweenness of node a are defined as follows:

$$\text{Betweenness, } C_B(a) = \sum_{i \neq j \neq a} \frac{\sigma(i, j|a)}{\sigma(i, j)} \tag{3.14}$$

$$\text{Normalized betweenness, } C'_B(a) = \frac{C_B(a)}{n_{Max}} = \frac{2}{(n-1)(n-2)} \sum_{i \neq j \neq a} \frac{\sigma(i, j|a)}{\sigma(i, j)} \tag{3.15}$$

• **Average Shortest Path Length**. A shortest path length, $d_{a,i}$ is the minimum distance (sum of edge weight) between a node a and other nodes i in the network. Average shortest

path length of a node a is the sum of shortest path length of node a averaged over the total number of shortest path of node a to other nodes (n_{sp}) and is defined as follows:

$$\text{Average shortest path length, } L(a) = \frac{1}{n_{sp}} \sum_{a \neq i} d_{a,i} = \frac{1}{n(n-1)} \sum_{a \neq i} d_{a,i} \tag{3.16}$$

• **Closeness**. Closeness is a measure of the efficiency of information flow in a network. It is the inverse of average shortest path length. A node with high closeness implies it is near on average topologically to other nodes. The closeness and normalized closeness of node a are given by:

$$\text{Closeness, } C_C(a) = \frac{1}{L(a)} = \frac{n(n-1)}{\sum_{a \neq i} d_{a,i}} \tag{3.17}$$

$$\text{Normalized closeness, } C'_C(a) = \frac{C_C(a)}{n-1} = \frac{n}{\sum_{a \neq i} d_{a,i}} \tag{3.18}$$

• **Clustering Coefficient**. Clustering is an important property of social networks: people tend to have friends who are also friends with each other. Clustering coefficient of a node a denoted by $CC(a)$ is a measure of the extent to which a node a in a graph tends to form cluster with other nodes. It is a real number between 0 (no clustering) and 1 (maximum clustering). One way to calculate $CC(a)$ is to check for triangles. That is, to check that when two edges share a node, the probability of a third edge exists such that the three edges form a triangle. A triplet is three nodes that are connected by either two (open triplet) or three (closed triplet) undirected edges. For example, $\boldsymbol{V} = \{a, b, d\}$ and $\boldsymbol{V} = \{b, c, d\}$ form two triangles as shown in Figure 3.5. Furthermore, $\boldsymbol{V} = \{a, b, c\}$ is an open triplet and $\boldsymbol{V} = \{a, b, d\}$ is a closed triplet. The local clustering coefficient of a node a and the global clustering coefficient are defined as follows (Opsahl & Panzarasa, 2009):

$$CC(a) = P(\text{two randomly selected neighbors of } a \text{ are neighbor}) \tag{3.19}$$

$$= \frac{3 \times \text{Number of triangles}}{\text{Number of all triplets}} \tag{3.20}$$

$$CC = \frac{1}{n} \sum_{i \in V} CC(i) \tag{3.21}$$

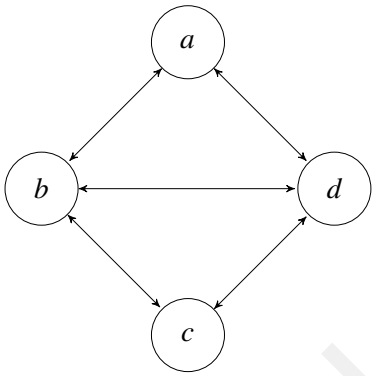

Figure 3.5: An example of triangles and triplets.

• **Neighbourhood Connectivity**. The connectivity of a node a denoted by $\kappa(a)$ is the minimum number of nodes whose deletion disconnects it (Maslov & Sneppen, 2002). A network with $\kappa(a) > 0$ is said to be connected (that is, there exists a path from node a to any other nodes in the network) and a graph with $\kappa = 0$ is said to be disconnected.

• **Edge Betweenness**. Betweenness centrality of an edge $e_{i,j}$ is the sum of the fraction of all-pairs shortest paths that pass through $e_{i,j}$ (Brandes, 2008). An edge with high edge betweenness centrality serves as a bridge-like connector between two parts of a network. The deletion of those high scored edges may affect the connectivity between many pairs of nodes through the shortest paths between them. We denote σ as the total number of shortest paths between all pairs of nodes and $\sigma(e_{i,j})$ as the number of those paths passing through edge $e_{i,j}$. The edge betweenness is defined as:

$$\text{Edge betweenness, } C_{EB}(e_{i,j}) = \sum_{i,j \in V} \frac{\sigma}{\sigma(e_{i,j})} \tag{3.22}$$

3.4.2 Shortest Path Analysis

Shortest path analysis highlights a route between any two nodes that requires a minimum sum of edge weights. Dijkstra's algorithm (Broumi et al., 2016) is one of the algorithms used to determine the shortest path and the corresponding path length from one source node to every other target node within the same graph, provided that the target nodes are reachable from the source node. Dijkstra's algorithm intends to create a shortest path tree from a single source node, by building a set of nodes that have minimum distance from the source. A shortest path tree is the set of edges connecting all nodes such that the sum of edge weights from the source to each target node is minimized.

The following terms are defined:

- Distance array, D is an array of minimum distances from the source node, s to each target node t in the graph.
- Unvisited array, U is a set of unvisited nodes.
- Visited array, V is a set of visited nodes.

The Dijkstra's algorithm is shown in Algorithm 3.1. At the end of the algorithm, U is empty and V contains all the nodes of the graph. All nodes in the graph are visited and the smallest distance to each node from the source node is found. Therefore, a shortest path tree is obtained.

Algorithm 3.1: Dijkstra's Shortest Path Algorithm
Input: The edge weight data, *w(i,j)* and a source node, s.
Output: The shortest paths of node s.
Run the following steps:
1: Determine U and V. At the beginning of the algorithm, V is empty and U consists of all nodes in the network.
2: Choose a source node, s. Assign $D(s) = 0$ for the source node (as the distance from node s to node s is 0) and $D(t) = \infty$ for all other nodes, t (as the actual minimum distance is unknown).
3: Check all the neighbors of the source node that are present in U in no specific order. Update $D(t)$ of each neighbor as follows: • If $D(s) + w(s,t) < D(t)$, there is a new minimal distance found for t, update $D(t)$ to the new minimal distance value $(D(s) + w(s,t))$; • Otherwise, no updates are made to $D(t)$.
4: Mark node s as visited after all neighbors of s are checked. Add the node to V and delete it from U.
5: Choose an unvisited node from U with the minimum $D(t)$ and denote it as a new source node.
6: Repeat step 3, 4 and 5 until U is empty.

3.5 Summary

We have presented some mathematical background on set theory, functions and probability theory. Moreover, we briefly discussed several analysis tools to evaluate our simulations, which include descriptive statistical analysis, social network analysis and shortest path analysis. These mathematical tools are the building blocks for the design of our schemes in this thesis.

CHAPTER 4: PROPOSED SCHEMES

In this chapter, we formalize two new privacy models to address the unlinkability component in social networks. With the two new models, we propose two secure anonymization schemes that satisfy both anonymity and unlinkability. The algorithm of the schemes are presented and the advantages and disadvantages of the schemes are further discussed in each section. The potential applications of the schemes are then followed.

4.1 Unlinkability in Weighted Social Networks

As shown in our gap analysis previously discussed in Chapter 2, most of the current work were built using the anonymity notion, where an attacker cannot sufficiently identify a target user from a graph. In this chapter, we proposed stronger approaches to anonymize social network data using unlinkability notion. Unlinkability of two or more objects of interest (for example, subjects, messages or actions) implies that an attacker cannot sufficiently distinguish whether these objects of interest are related or not within a communication channel, even if the source and destination can each be identified as participating in the channel (Lee et al., 2014; Pfitzmann & Hansen, 2010; Thiel et al., 2013; Zhuang et al., 2005). For example, given a scenario with at least two senders, two messages sent are unlinkable to an adversary, if the probability that these two messages are sent by the same sender is sufficiently close to 1/(number of senders). Unlinkability is often supported by the use of misinformation (inaccurate or erroneous information, provided usually without conscious effort at misleading, deceiving, or persuading one way or another) or disinformation (deliberately false or distorted information given out in order to mislead or deceive) to lead to a growing uncertainty of the attacker regarding which information is correct (Pfitzmann & Hansen, 2010). The above definition is used in a communication system whereas in the present work, a social network data publication is considered. In

this section, we adopt the definition of unlinkability to graph data publishing scenarios.

4.1.1 Notation

Before we define the *edge weight unlinkability* and *node unlinkability* notions, we present the definition of some key terms and notation used in our work, as shown in Table 4.1.

Table 4.1: Notation.

Symbol	Meaning
m	Number of edges in the network
n	Number of nodes in the network
$\boldsymbol{E}$	Set of edges
$\boldsymbol{V}$	Set of nodes
G	Original graph data
G'	Published graph data
$\boldsymbol{W}$	Weight sequence (Sequence of weight in ascending order)
$\boldsymbol{W'}$	Perturbed weight sequence
$\boldsymbol{W(a)}$	Set of edge weights associated with node a
$\boldsymbol{W(a \cup b)}$	Set of edge weights associated with node a and node b
$W(a,b)$	Edge weight from node a to node b
$W'(a,b)$	Perturbed edge weight from node a to node b
$w_{i,j}$	Edge weight between node i and j
w_p	Edge weight in weight sequence for p =1, 2, 3, ..., m
w'_p	Perturbed edge weight in weight sequence for p =1, 2, 3, ..., m
Z_T	Universal set (Set of distinct values of $\boldsymbol{W}$)
$N(Z_T)$	Complete frequency set (Set of frequency of distinct values in $\boldsymbol{W}$)
Z_p	Possible set (Set of values that satisfy *edge weight unlinkability*)
$N(Z_p)$	Frequency set (Set of frequency of distinct values in the possible set)
$\boldsymbol{S}(a,b)$	Candidate set (Set of values that satisfy *node unlinkability*)
N	Number of distinct edge weight values
m_{Add}	Number of fake edges added
n_{Add}	Number of fake nodes added

4.1.2 Edge Weight Unlinkability

We define *edge weight unlinkability* as below.

Definition 1. ***Edge weight unlinkability***

Given an edge weight $w \in \boldsymbol{W}$ with value X in an original network G, w is said to be unlinkable if w is perturbed to w' with value Y in the published network G', where $X \neq Y$ and there does not exist an injective function: $f(Y) \mapsto X$ that maps value Y in the published data to value X in the original data. An anonymized data is said to be edge weight unlinkable if all edge weights in perturbed network G' satisfy *edge weight unlinkability* such that the perturbed edge weight value does not equal to its original edge weight value for all edge weight in weight sequence and there does not exist an injective function f between the original and published data. In mathematical notation, $w_{i,j} \neq w'_{i,j}$, $\forall w_{i,j} \in \boldsymbol{W}, \forall w'_{i,j} \in \boldsymbol{W'}, \forall i, j \in \boldsymbol{V}$ and $f(Y) \mapsto X$ is not an injective function.

Edge weight unlinkability prevents the inference of true edge weights of a user and satisfies two essential properties:

1. All original edge weights are modified such that $w \neq w'$. If an edge weight value of a user is not modified, then a user is still linkable to that piece of information.
2. There is no injective mapping from the original edge weight to the perturbed edge weight, such that the original edge weight cannot be reverse-engineered to draw a defined estimation about the original edge weight.

4.1.3 Node Unlinkability

We define *node unlinkability* as below.

Definition 2. ***Node unlinkability***

Given a node a with associated edge weight sequence $\boldsymbol{W(a)}$ in an original network G and $\boldsymbol{W'(a)}$ in the published network G', the node is said to be unlinkable if $\forall w \in \boldsymbol{W(a)} \Leftrightarrow$

$\nexists w \in \boldsymbol{W'(a)}$ and there does not exist an injective function f mapping the original value X to new value Y. An anonymized data is said to be node unlinkable if all nodes in G' satisfy *node unlinkability* such that $\forall v \in \boldsymbol{V} \wedge \forall w \in \boldsymbol{W(v)} \Rightarrow \nexists w \in \boldsymbol{W'(v)}$ and $f(Y) \mapsto X$ that maps value Y in the published data to value X in the original data is not an injective function.

Node unlinkability prevents the linkability of edge weight information to its associated users in the original data. Thus, an adversary could not link the auxiliary edge weight information to intrude the identity of a user using a linkage attack.

4.1.4 Discussion

In this discussion, we compare and establish relationship between the two proposed privacy models. We show that *node unlinkability* implies *edge weight unlinkability* but not vice versa, as presented in proposition 1 and 2. Furthermore, we show that any data satisfying *node unlinkability* are invulnerable against edge weight disclosure and identity disclosure. To be precise, we prove that there does not exist an injective mapping function that links associated edge weights to its corresponding node in the perturbed data as shown in proposition 3. Moreover, no linkage attack is possible to reidentify a target node in the published data using edge weight information as background knowledge, as proven in proposition 4. Unlinkability is a sufficient condition of anonymity, but it is not a necessary condition (Pfitzmann & Hansen, 2010). That is, unlinkability implies anonymity. Hence, the proposed definitions fulfill both unlinkability and anonymity components.

Proposition 1. *Node unlinkability implies edge weight unlinkability.*

Proof. From the definition of *node unlinkability*, $\forall w \in \boldsymbol{W(a)} \Leftrightarrow \nexists w \in \boldsymbol{W'(a)}$. Since the selection of new edge weight from W is mutually exclusive and exhaustive, $\boldsymbol{W(a) \neq W'(a)}$ as

shown in Figure 4.1. Thus, $\forall w_{i,j} \in \boldsymbol{W(a)} \wedge \forall w'_{i,j} \in \boldsymbol{W'(a)} \Rightarrow w_{i,j} \neq w'_{i,j}$. This completes the proof. □

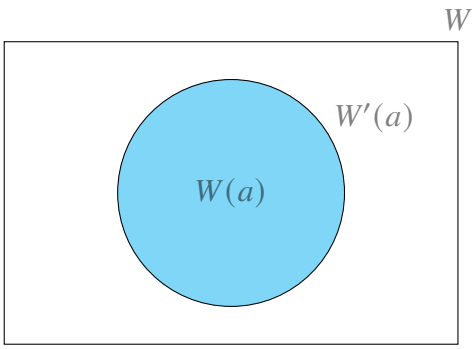

Figure 4.1: A venn diagram of edge weight.

Proposition 2. *Edge weight unlinkability does not imply node unlinkability.*

Proof. Using a counterexample as shown in Figure 4.2, we show that $w_{i,j} \neq w'_{i,j}, \forall v \in \boldsymbol{V}$. Thus, *edge weight unlinkability* is satisfied. However, $\forall w \in \boldsymbol{W(1)} \Leftrightarrow \exists w \in \boldsymbol{W'(1)}$. For example, edge weight value 12 is still connected to node 1. Hence, *node unlinkability* is not satisfied. This completes the proof. □

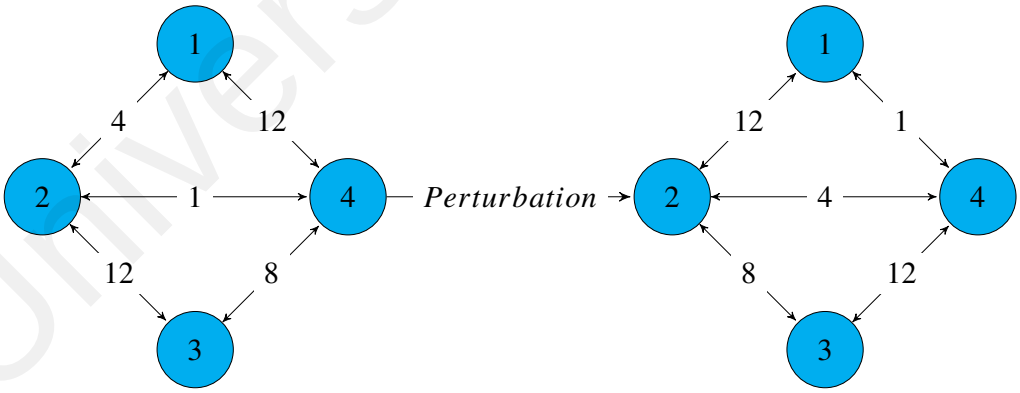

Figure 4.2: A counterexample.

Proposition 3. *Given there exists a function g that maps a set of edge weight,* ***W(a)*** *to a node* ***a*** *in original data, such function g does not exist in a perturbed data that satisfy node unlinkability.*

Proof. We prove by contradiction. Given that $\forall w \in \boldsymbol{W(a)}$ is associated (mapped) to a node $a \in \boldsymbol{V}$, we have a function g such that $g(w) \mapsto a$. This existence of the function g indicates that node a is associated with some edge weights w. First, we assume that such function g exists in the perturbed data $\boldsymbol{W'(a)}$. However, based on the definition of *node unlinkability*, $\forall w \in \boldsymbol{W(a)} \Rightarrow \forall w \notin \boldsymbol{W'(a)}$, we know that there does not exist a function $g(w)$ that maps $w \in \boldsymbol{W(a)}$ to the node a in the perturbed data as all the associated edge weights of node a are modified such that $w \notin \boldsymbol{W'(a)}$. Here, we have arrived at a contradiction where our original assumption (function g exists in a perturbed data that satisfy *node unlinkability*) could not be true. This completes the proof. □

Proposition 4. *Given an adversary possesses a complete edge weight information of a known target node* $\boldsymbol{a}$ *that exists in the network, the adversary fails to reidentify correctly node* $\boldsymbol{a}$ *in the published data that satisfy node unlinkability using a linkage attack.*

Proof. There are only three possible outcomes of the reidentification. Let b denotes as an arbitrary node in the network and $\boldsymbol{W'(b)}$ is the associated edge weight of b that are published.

Outcome 1 : There is no exact match of $\boldsymbol{W(a)}$ and $\boldsymbol{W'(b)}$. Thus, $\forall a, b \in \boldsymbol{V} \ni \boldsymbol{W(a)} \neq \boldsymbol{W'(b)}$.

$\therefore$ No identity is inferred from the published data.

Outcome 2 : There is at least one exact match of $\boldsymbol{W(a)}$ and $\boldsymbol{W'(b)}$. We have $\forall a, b \in \boldsymbol{V} \ni \exists \boldsymbol{W'(b)} = \boldsymbol{W(a)}$. From the definition of *node unlinkability*, $\boldsymbol{W(a)} \neq \boldsymbol{W'(a)}$. This implies that $\boldsymbol{W'(b)} \neq \boldsymbol{W'(a)}$. However, it can be deduced that: $a = b \Rightarrow \boldsymbol{W(a)} = \boldsymbol{W(b)} \Rightarrow \boldsymbol{W'(a)} = \boldsymbol{W'(b)}$. Hence, $\boldsymbol{W'(b)} \neq \boldsymbol{W'(a)} \Rightarrow b \neq a$.

$\therefore$ Although there is an exact match, a is not the true identity of node b.

Outcome 3 : There is at least one partial match of $\boldsymbol{W(a)}$ and $\boldsymbol{W'(b)}$. Thus, $\forall w \in \boldsymbol{W(a)}$, $\forall w' \in \boldsymbol{W'(b)} \Rightarrow \exists w = w'$. However, from *node unlinkability*, we have $\forall w \in \boldsymbol{W(a)} \Rightarrow \forall w \notin$

$\boldsymbol{W'(a)}$, which implies that w must not be an edge weight of node a in the published data. Hence, if $w \in \boldsymbol{W(a)}$ is an edge weight of node b in the published graph, then node a and b must not be the same individual.

$\therefore$ Node a cannot be reidentified by linking the edge weight information to the published data.

Therefore, although an adversary has the complete edge weight data of a known target node a, the adversary fails to correctly reidentify node b from the published data using a linkage attack. This completes the proof. □

4.2 *MinSwap*

In this section, we design ***MinSwap*** which deploys *edge weight unlinkability* model that modifies the edge weights using perturbation (Rodriguez-Garcia et al., 2019). The modification is based on the idea of data swapping to preserve the edge weight distribution and therefore its statistical properties. The *MinSwap* algorithm is presented and its applications are discussed.

4.2.1 *MinSwap* Algorithm

MinSwap consists of two main phases, namely possible set determination and candidate selection. The edge weight data is perturbed by exchanging edge weight values among data tuples to achieve privacy preservation. Data swapping is a value-invariant method, where edge weight distribution is not altered by the program execution but the edge weight sequence is altered. Data swapping preserves the univariate statistics such as mean, variance, distribution and lower-order multivariate statistics such as covariance reasonably. A pseudo algorithm of *MinSwap* is presented in Algorithm 4.1.

Algorithm 4.1: Minimal Swapping Strategy (*MinSwap*)	
Input: The original edge weight sequence, $\boldsymbol{W}$	
Output: The perturbed edge weight sequence, $\boldsymbol{W'}$	
1	Find Z_T and $N(Z_T)$.
2	**for** p from 1 to m,
3	{Find Z_p and $N(Z_p)$.
4	**if** $N(Z_p) \neq \emptyset$, **then**
5	{Calculate $Prox(w)$ for each $w \in Z_p$.
6	Determine max $Prox(w)$.
7	Find corresponding w.
8	Update $N(Z_T)$. }
9	**else**
10	{Select a value w from Z_p randomly.
11	Record w in $U(Z_T)$. }
12	Assign the value w to w'_p.}
13	**return** $\boldsymbol{W'}$.

An **edge weight sequence**, denoted by $\boldsymbol{W}$ is the sequence of all edge weight of an original network in ascending order. We denote Z_T as the **universal set** containing all distinct values of $\boldsymbol{W}$, $N(Z_T)$ as the **complete frequency set** recording the frequency of values in $\boldsymbol{W}$, Z_p as the **possible set** of w_p containing all values that satisfy *edge weight unlinkability* and $N(Z_p)$ as the **frequency set** of w_p recording the frequency of values in the possible set. Each phase of Algorithm 4.1 is explained as below:

• **Possible Set Determination (line 1-4 in Algorithm 4.1).** An edge weight sequence is added as an input database to Algorithm 4.1. In the first phase, possible candidates that satisfy *edge weight unlinkability* are determined from the original data, $\boldsymbol{W}$. The Z_T and $N(Z_T)$ are determined to learn the frequency distribution of the input database. Then, the possible set Z_p and $N(Z_p)$ of an edge weight w_p are determined such that $Z_p = Z_T - \{w_p\}$. Here, the new edge weight (qualified candidate) is selected from the possible set Z_p so that the anonymized data satisfy *edge weight unlinkability*.

• **Candidate Selection (line 5-8 in Algorithm 4.1).** New edge weight, w' is selected from Z_p based on the maximum of the proximity function, which we define as:

$$Prox(w) = \frac{\text{Frequency of } w \text{ in } Z_p}{|w_p - w|}, \forall w \in Z_p \tag{4.1}$$

This function serves two purposes: it allows a nearer value to be selected (a lower information loss) and over the iterations in greedy algorithm 4.1, one value could be mapped to different new values (injective function does not exist). This increases the uncertainty of an adversary in inferring the original edge weight value. In the end of this phase, the selected value w is assigned to w'_p and the corresponding frequency of w in $N(Z_T)$ is updated.

• **Special Case (line 10-11 in Algorithm 4.1).** $N(Z_p) = \emptyset$ implies that all values from the possible set are completely consumed. In this case, w' is selected from Z_p randomly, imposing a certain amount of distortion to the original data distribution. However, randomness is applied to provide a higher privacy protection. $U(Z_T)$ is utilized to record the frequency of the overused w. This scenario only occurs when there is a dominant value in the original data (> 50% of the edge weight data). **Nevertheless, the existence of a solution for Algorithm 1 is guaranteed, regardless of the type of distribution of original data.**

4.2.2 Discussion

An example of *MinSwap* is demonstrated in Table 4.2 using data in Figure 1.2. At first iteration, the possible set Z_1 for $w_1 = 1$ is {2, 4, 8, 10, 12, 14, 15} and the corresponding frequency set $N(Z_1)$ is {-, 1, 1, 2, 4, 1, 1, 1} (which is obtained by referring the corresponding frequency of each $w \in Z_1$ in $N(Z_T)$). Hence, new edge weight w'_1 is 2, according to the corresponding max $Prox(w)$. The frequency of 2 is reduced by 1 in the

Table 4.2: An example of *MinSwap*.

p	$\boldsymbol{W}$	$N(Z_p)$								$\boldsymbol{W'}$
Z_T		1	2	4	8	10	12	14	15	
$N(Z_T)$		1	1	1	2	4	1	1	1	
1	1	-	1	1	2	4	1	1	1	2
2	2	1	-	1	2	4	1	1	1	1
3	4	0	0	-	2	4	1	1	1	10
4	8	0	0	1	-	3	1	1	1	10
5	8	0	0	1	-	2	1	1	1	10
6	10	0	0	1	2	-	1	1	1	8
7	10	0	0	1	1	-	1	1	1	8
8	10	0	0	1	0	-	1	1	1	12
9	10	0	0	1	0	-	0	1	1	14
10	12	0	0	1	0	1	-	0	1	10
11	14	0	0	1	0	0	0	-	1	15
12	15	0	0	1	0	0	0	0	-	4
Final $N(Z_T)$		0	0	0	0	0	0	0	0	

$N(Z_T)$. At the end of algorithm, the final $N(Z_T) = 0$ shows that all the original data are inter-swapped with each other and thus the original distribution is fully preserved.

We show that *MinSwap* fulfills *edge weight unlinkability* in proposition 5.

Proposition 5. Anonymized data after *MinSwap* satisfy *edge weight unlinkability*.

Proof. From Definition 1, $\forall w_p \in \boldsymbol{W}, \forall\, w'_p \in \boldsymbol{W'}, \forall p \in [1,m] \Rightarrow w_p \neq w'_p$. The new edge weight is selected from Z_p and the selection of w'_p is mutually exclusive event. Hence, $w'_p \in Z_p \Rightarrow w'_p \notin [Z_p]^c \Rightarrow w'_p \notin \{W_p\} \Rightarrow w'_p \neq w_p$. This completes the proof. □

It is not possible to reverse engineer and discover the true edge weight using a linkage attack as there does not exist an injective mapping from the published data and the original data. From the utility aspect, the statistical properties of edge weight data are preserved as the anonymized data is a permutated version of the original data. This is a scheme designed for networks where the identity of nodes are public knowledge but the edge weight values are sensitive information. No structural anonymization is required to protect

the identity of nodes and thus more utility could be preserved. Examples of such networks include research communities (ResearchGate and DBLP) and professional sites (LinkedIn and JobStreet).

4.3 δ-*MinSwapX*

In this section, we design another scheme namely δ-***MinSwapX*** based on *node unlinkability* to address edge weight disclosure, link disclosure and node reidentification simultaneously with a minimal data utility trade-off. This scheme consists of edge weight modification using perturbation and structural modification using randomization.

4.3.1 Edge Weight Modification

Perturbation is deployed to prevent edge weight disclosure and node reidentification using edge weight data as the background knowledge. It consists of two main phases, namely candidate set determination and minimal candidate selection.

4.3.1.1 Algorithm

Each phase of edge weight modification is presented as follows:

Algorithm 4.2: Candidate Set Determination	
1	Find universal set Z_T.
2	Find set $\boldsymbol{W(a)} = \{W(a,b) \mid \forall\, a,b \in [1,n]\}$.
3	Find set $\boldsymbol{W(a \cup b)} = \boldsymbol{W(a)} \cup \boldsymbol{W(b)}$.
4	Find candidate set, $\boldsymbol{S} = \{s \mid s \in Z_T - \boldsymbol{W(a \cup b)}\}$.

• **Candidate Set Determination (Algorithm 4.2).** The universal set that contains all the edge weight values (Z_T) is separated into two mutually exclusive sets, namely candidate set ($\boldsymbol{S}$) and associated edge weight set ($\boldsymbol{W(a \cup b)}$). A **candidate set** is the set that collects all the possible candidates such that the candidate $s \in \boldsymbol{S}$ is not associated with node a and b. A candidate set is given by $\boldsymbol{S}(a,b) = \{s \mid s \in Z_T - \boldsymbol{W(a \cup b)}\} = Z_T \setminus \boldsymbol{W(a \cup b)}$. This is to

Algorithm 4.3: Edge Weight Modification	
Input: The original edge weight data, $\boldsymbol{W}$	
Output: The perturbed edge weight data, $\boldsymbol{W'}$	
1	Determine the weight sequence, $\boldsymbol{W}$.
2	Find candidate sets for all edge weights.
3	**for** $p = 1$ to m
4	{**call** algorithm 2 to determine the candidate set, $\boldsymbol{S}$.
5	Assign $w'_p = \min \lvert s - w_p \rvert + w_p$, for $\forall s \in \boldsymbol{S}$. }
6	**return** $\boldsymbol{W'}$.

ensure that $\boldsymbol{S}$ contains all the qualified candidates that satisfy *edge weight unlinkability* and *node unlinkability*, as shown in proposition 6.

• **Minimal Candidate Selection (line 5 in Algorithm 4.3).** A candidate is selected based on the least value change to guarantee minimum information loss, as shown in proposition 7.

Proposition 6. *Anonymized edge weight data post-implementation of Algorithm 4.3 satisfy node unlinkability.*

Proof. From the definition 2, we have $\forall w \in \boldsymbol{W(a)} \Rightarrow \nexists w \in \boldsymbol{W'(a)} \Rightarrow \forall w \notin \boldsymbol{W'(a)}$. Given that $Z_T = \boldsymbol{S} \cup \boldsymbol{W(a \cup b)}$, this implies $\forall w \in \boldsymbol{W(a)} \Rightarrow \forall w \notin \boldsymbol{S}$. Since the new edge weight is selected from $\boldsymbol{S}$ only, we have $w' \in \boldsymbol{W'(a)} \subseteq \boldsymbol{S}$, which means that $\forall w \notin \boldsymbol{S} \Rightarrow \forall w \notin \boldsymbol{W'(a)}$. $\therefore \forall w \in \boldsymbol{W(a)} \Rightarrow \forall w \notin \boldsymbol{W'(a)}$. Hence, *node unlinkability* is satisfied, which further implies *edge weight unlinkability*. This completes the proof. □

Proposition 7. *The information loss due to Algorithm 4.3 is minimum.*

Proof. The information loss occurs during minimal candidate selection. At each iteration, the information loss is $\lvert w'_p - w_p \rvert$. This is the noise injected. The total information loss is $\sum_{p=1}^{m} \lvert w'_p - w_p \rvert$, where m is the number of original data. Since w'_p is selected based on

the lowest value change (min $|s - w_p|$), the total information loss due to Algorithm 3 is minimum. This completes the proof. □

4.3.1.2 Discussion

Table 4.3: An example of Algorithm 4.2 and 4.3.

p	$\boldsymbol{W}$	Value	$\boldsymbol{W}(a \cup b)$ (value)	$\boldsymbol{S}$ (value)	$\boldsymbol{W'}$
1	$W(2,4)$	1	1, 4, 8, 10, 14, 15	2, 12	2
2	$W(6,7)$	2	2, 8, 12, 14, 15	1, 4, 10	1
3	$W(1,2)$	4	1, 4, 8, 10, 14	2, 12, 15	2
4	$W(2,8)$	8	1, 4, 8, 10, 12, 14	2, 15	2
5	$W(3,7)$	8	2, 8, 10, 15	1, 4, 12, 14	4
6	$W(5,8)$	10	8, 10, 12	1, 2, 4, 14, 15	14
7	$W(1,4)$	10	1, 4, 10, 15	2, 8, 12, 14	8
8	$W(2,5)$	10	1, 4, 8, 10, 14	2, 12, 15	12
9	$W(3,8)$	10	8, 10, 12	1, 2, 4, 14, 15	14
10	$W(6,8)$	12	2, 8, 10, 12, 14	1, 4, 15	15
11	$W(2,6)$	14	1, 2, 4, 8, 10, 12, 14	15	15
12	$W(4,7)$	15	1, 2, 8, 10 ,15	4, 12, 14	14

Using the same data set from Figure 1.2, an example of Algorithm 4.3 and 4.4 is demonstrated in Table 4.3. At first iteration, $\boldsymbol{W}(2 \cup 4) = \boldsymbol{W}(2) \cup \boldsymbol{W}(4) = \{1, 4, 8, 10, 14\} \cup \{1, 10, 15\} = \{1, 4, 8, 10, 14, 15\}$. Hence, $\boldsymbol{S} = \{1, 2, 4, 8, 10, 12, 14, 15\} \setminus \boldsymbol{W}(2 \cup 4) = \{2, 12\}$ and $w'_1 = (2 - 1) + 1 = 2$. The iterations terminate at $p = m = 12$.

The perturbed data satisfy both *edge weight unlinkability* and *node unlinkability*. A user could not be retraced using edge weight data of the targeted victim as the associations between the edge weights and the nodes have been broken completely. From the utility perspective, we have minimally changed the data so that no excessive utility is loss due to the edge weight modification. If there does not exist a candidate set for a particular edge weight, then no new edge weight is published for that particular edge weight to secure the privacy of a user. However, this is not common in a scalable network which contains high diversity of edge weight values.

4.3.2 Structural Modification

Randomization is deployed to modify the network structure to prevent node reidentification using structural data as background knowledge and to prevent link disclosure. It consists of four phases, namely edge deletion, fake node addition, fake edge addition and edge weight addition.

4.3.2.1 Algorithm

A pseudo algorithm for structural modification is presented in Algorithm 4.4. Each phase in the structural modification is elaborated as follows:

• **Edge Deletion (line 1-6 in Algorithm 4.4).** Edge betweenness centrality is calculated to determine the most influential nodes and edges in the network. **Edge betweenness** is the number of shortest paths between pairs of nodes that run along an edge. An edge should not be removed if the edge is important in the network (high edge betweenness). A user-defined parameter δ is selected to remove δ of the existing edges in the ascending order of edge betweenness (line 2-5 in Algorithm 4.4). We denote a set $\boldsymbol{C}$ as a checker to record the associated nodes where its edges remain intact post edge deletion.

• **Fake Node Addition (line 7 in Algorithm 4.4).** A minimal number of fake nodes d are added into the network to hide the existence of a target victim in the anonymized data. The number of fake nodes, n_{Add} is calculated as:

$$n_{Add} = max(\lceil \frac{|\boldsymbol{C}|}{D_{mode}} \rceil, 1) \tag{4.2}$$

where $|\boldsymbol{C}|$ is the size of $\boldsymbol{C}$, D_{mode} is the mode of degree and $\lceil \frac{|\boldsymbol{C}|}{D_{mode}} \rceil$ is the least integer greater than or equal to $\frac{|\boldsymbol{C}|}{D_{mode}}$. If there are at least one mode, a larger mode is selected. By considering the mode of degree (degree that appears most often) in original network, all

Algorithm 4.4: Structural Modification	
Input: The perturbed edge weight data, $W'(a,b)$	
Output: Perturbed data that resist edge weight disclosure, link disclosure and identity disclosure	
1	Define a parameter, δ, where $0 \leq \delta \leq 1$.
2	**if** $\delta = 0$, **then** exit the algorithm.
3	**else**
4	{Edge betweenness is calculated for each edge using original edge weight data. Denotes $\boldsymbol{C}$ as a checker set containing all nodes in the network.
5	Remove δ of the existing edges according to the ascending order of edge betweenness.
6	Record the edge (a, b) that has been removed.
	* Remove the corresponding nodes a and b from $\boldsymbol{C}$.
7	Add n_{Add} fake nodes d into the network.
	* $n_{Add} = max(\lceil \frac{\lvert \boldsymbol{C} \rvert}{D_{mode}} \rceil, 1)$, where D_{mode} is mode of degree. If there are at least one mode, choose maximum mode.
8	**while** $\boldsymbol{C} \neq \emptyset$,
9	{Add edges between the remaining nodes c in $\boldsymbol{C}$ and the fake nodes d randomly.
	* Randomly select D_{mode} of the remaining nodes c from $\boldsymbol{C}$ to form edges with the fake nodes d.
10	Record the edge (c, d) that has been formed.
	* Remove the corresponding nodes c from $\boldsymbol{C}$.
	}
11	**for** each inserted fake edge, assign an edge weight value w' to the edge,
12	{**if** $\exists w \in \boldsymbol{S(c)} \ni w > Max[w(c)]$, **then** $w' = min\ w$.
13	**else** $w' = Max[\boldsymbol{S(c)}]$. } }
14	**return** perturbed data.

the fake nodes are likely to possess approximately the same degree as the majority nodes in the network (the presence of a fake node is hidden). Furthermore, important nodes are preserved in the anonymized network as no true node is removed from the network.

• **Fake Edge Addition (line 8-10 in Algorithm 4.4).** A D_{mode} number of remaining nodes c are selected randomly from $\boldsymbol{C}$ to form edges with the fake nodes d (line 8-10). The total number of fake edge added denoted by m_{Add} is $|\boldsymbol{C}|$. Set $\boldsymbol{C}$ records the change of structural information. If a node's degree has changed, the node is removed from $\boldsymbol{C}$. An

empty $\boldsymbol{C}$ indicates that all nodes' degree are changed. Due to the randomness property of the newly added edges, an adversary could not confidently infer the structural properties of the target victim from the published graph. In fact, all structural information are modified in the output of the Algorithm 4.4. Furthermore, the structure of the graph is changed without compromising the important nodes and edges in the original network.

• **Edge Weight Addition (line 11-13 in Algorithm 4.4).** A new weight is inserted to each fake edge, which is selected from candidate set of the original node c so that it satisfies *node unlinkability*, such that $w' \in \boldsymbol{S(c)}$. Furthermore, to minimize the influence of these fake edges on the shortest paths of the original network, the new edge weight must satisfy one of the following conditions:

1. If there exists a set of values such that $w \in \boldsymbol{S(c)} \ni w > Max[w(c)]$, then $w' = min\ w$.
2. Else, $w' = Max[\boldsymbol{S(c)}]$.

The pseudo algorithm of δ-*MinSwapX* is a combination of Algorithm 4.2, 4.3 and 4.4. An example of Algorithm 4.4 is demonstrated using the same data set from Figure 1.2. Figure 1.2 and 4.3 show the network before and after edge weight modification while Figure 4.4, 4.5 and 4.6 show the network representations after each phase in structural modification.

4.3.2.2 Discussion

The overall edge modification algorithm is flexible and random. During the edge deletion process, a parameter δ is defined to determine which edges in the network should be removed. Important edges could be preserved as the edges are deleted according to the influence of edge (edge betweenness). During the edge addition process, the new edges are randomly inserted between the fake nodes and existing nodes in the original network

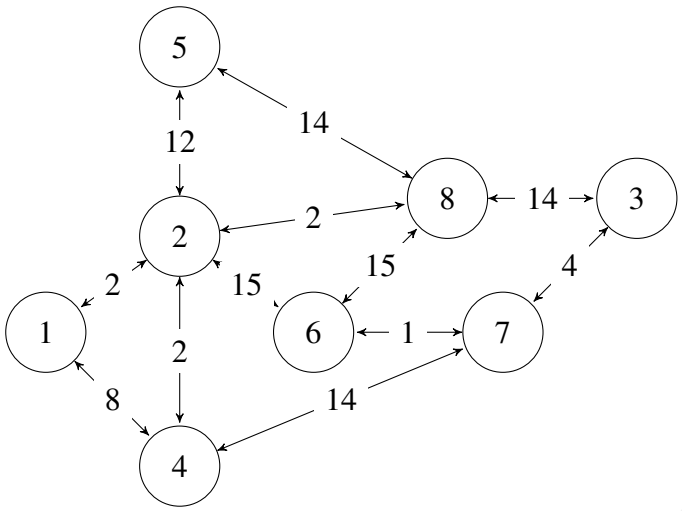

Figure 4.3: Original network after edge weight modification.

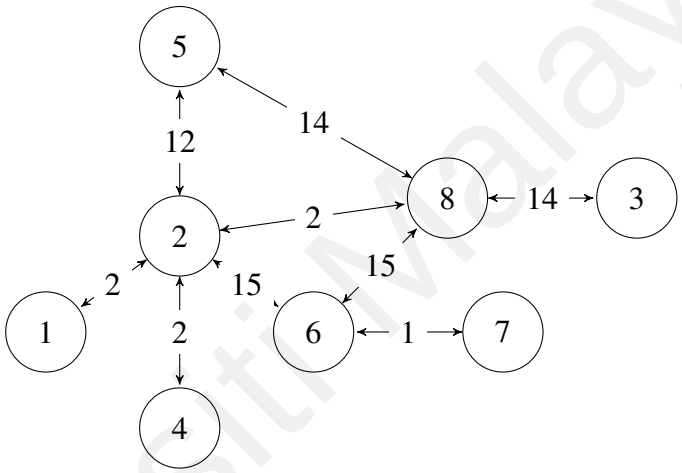

Figure 4.4: Network after edge deletion.

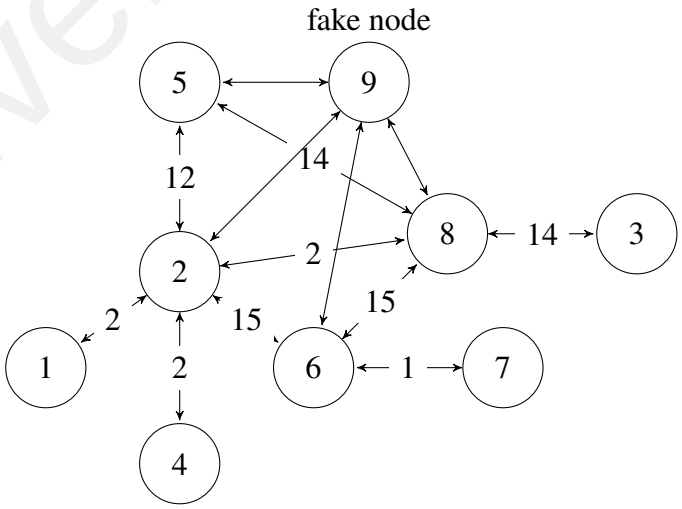

Figure 4.5: Network after fake node and edge addition.

to hide the true nodes and edges. The δ is used to control the balance between privacy level and utility level. Higher value of δ implies more deletions of true link and thus,

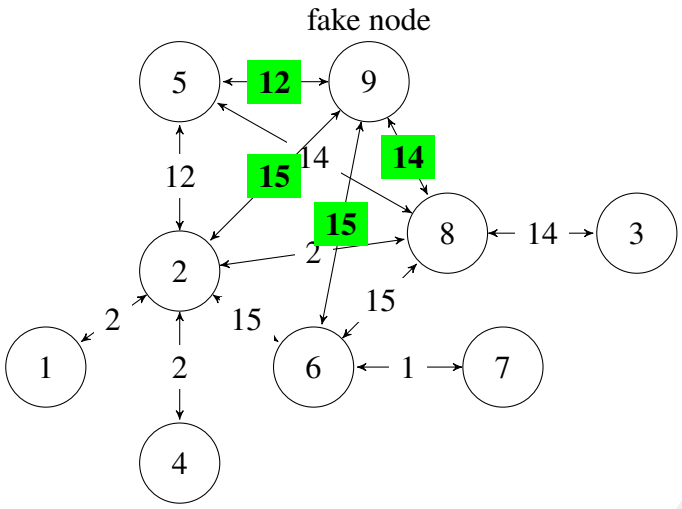

Figure 4.6: Network after edge weight addition.

the probability of link disclosure is reduced. However, this implies larger amount of distortion on the network structure.

Regardless of the δ value defined, the structural information of all real nodes are modified post-implementation of δ-*MinSwapX*. In addition, the edge weight value of the fake edges do not affect the shortest path in original network as the assigned values are slightly larger or equal to the edge weight involved in that particular shortest path. Hence, the background knowledge of an adversary cannot be utilized to map to the published data for node reidentification as the edge weight and structural information are unlinkable and randomized.

We assume the parameter δ is available to both data miners and attackers (Hay et al., 2007; Ying & Wu, 2008). The reason is that δ denotes the magnitude of randomization which may be needed to analyze the perturbed graph by data miners. Although δ is known, the identity and link of a user is still protected through the edge randomization process. Note that if $\delta = 1$, the published graph is a null graph (graph with no edge) with $n + 1$ nodes, which clearly contains almost no information about the original graph. We intend to have δ to be a small value.

This is a scheme designed for networks where the identity, the links and the edge weight data of a user are sensitive information. Edge weight anonymization and structural anonymization are applied simultaneously to fully protect a network user. Examples of such networks include healthcare networks (Doctor On Demand, HelloMD and LiveHealth Online) and social media networks (Facebook, Twitter and Instagram).

4.4 Summary

In this chapter, we addressed the problems of preserving privacy in publishing weighted network data. We discussed the unlinkability component of social networks and formalized two new unlinkability notions in weighted networks, namely *edge weight unlinkability* and *node unlinkability*. Security proof on the proposed models are then followed. With the formalized privacy models, we designed two new anonymization schemes, namely *MinSwap* and δ-*MinSwapX* for a secure and useful sharing of network.

CHAPTER 5: SECURITY AND PERFORMANCE ANALYSIS

In this chapter, we evaluate the performance of our schemes on three scalable real data sets. The experiments were conducted on a machine running Microsoft Windows 10 Home Single Language operating system, with an Intel Core TM i7- 8750H 2.20 GHz CPU and 16GB RAM. The algorithms were implemented in Python 3.7. Cytoscape 3.7.2 was used to analyze the network centrality of the data sets. Experimental results on real data sets show that our schemes efficiently achieve data utility preservation and privacy protection simultaneously.

5.1 Data sets

Three real data sets are used in the experiments to study the effects of our schemes on the data quality in terms of security, efficiency and utility. We extracted a subset of ***Bitcoin Alpha***[1], ***Facebook Artist***[2] and ***Youtube***[3] to validate the proposed schemes. All the data considered were weighted and non-directed. The details of the data sets are shown in Table 5.1.

5.2 Security Evaluation

The proposed $MinSwap$ and δ-*MinSwapX* guarantee that there is no injective function between the original database and the anonymized database. The scatter plots of new value versus original value of each dataset under both schemes are shown in Figures 5.1, 5.2, 5.3, 5.4, 5.5 and 5.6. These figures demonstrated that there is no injective mapping between the original data and the anonymized data, such that when $f(x_1) = f(x_2) \Rightarrow x_1 \neq x_2$ for all values.

[1] http://snap.stanford.edu/data/soc-sign-bitcoin-alpha.html

[2] https://snap.stanford.edu/data/gemsec-Facebook.html

[3] https://snap.stanford.edu/data/com-Youtube.html

Table 5.1: Description of the data sets.

Data Set	Nodes	Edges	Details
Bitcoin Alpha	3320	10554	Bitcoin is a peer-to-peer payment system without central authority. Bitcoin Alpha is a platform network which allows users to trade using Bitcoin. In this network, Bitcoin users are anonymous, but users' reputation score are required to reflect the reliability of the traders. Nodes represent the traders, edges are bitcoin transactions and edge weights are rating towards other traders.
Facebook Artist	50515	819306	Facebook is an online social network which allows its users to comment, share photo, post links, chat live and watch video. The data represent mutual like network among verified Facebook pages of artist category and were collected in 2017. Nodes represent the pages, edges are mutual likes among them and edge weights are the number of mutual likes.
Youtube	368548	1048572	Youtube is a video-sharing network, where users represent nodes and they can form friendship with other users in a group. Edge weights are the number of mutual likes.

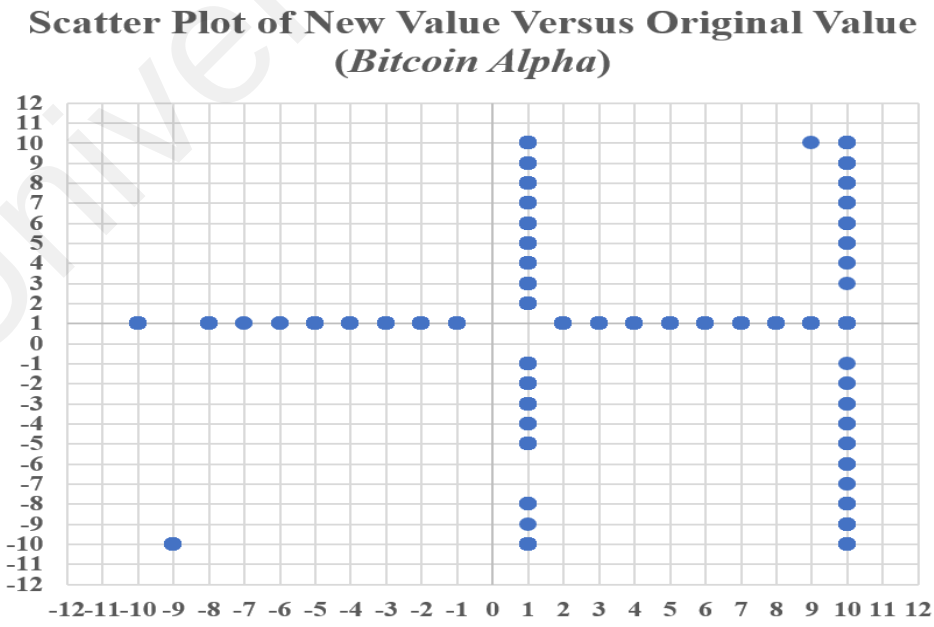

Figure 5.1: Scatter plots of new value versus original value in *Bitcoin Alpha* (*MinSwap*).

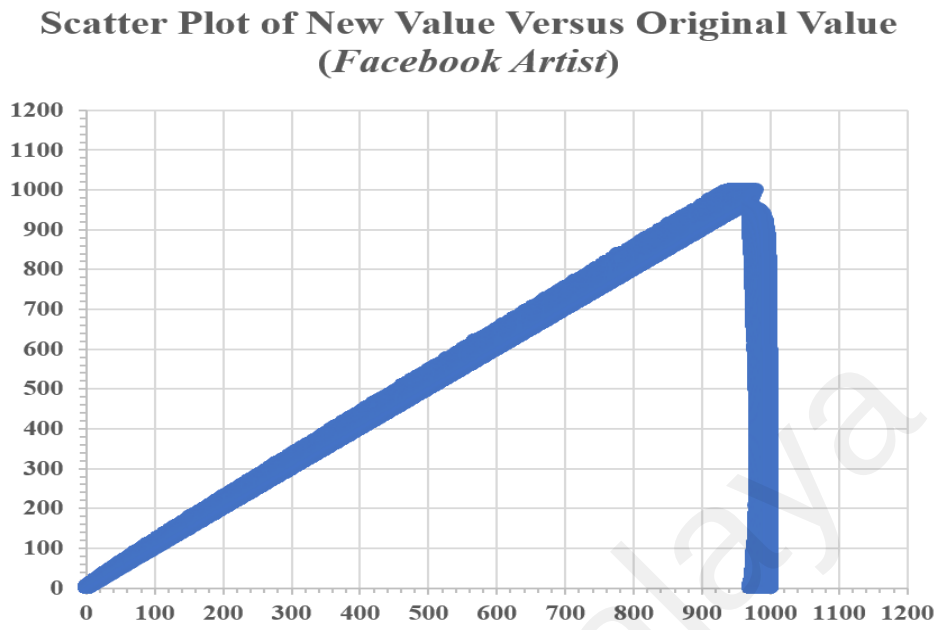

Figure 5.2: Scatter plots of new value versus original value in *Facebook Artist* (*MinSwap*).

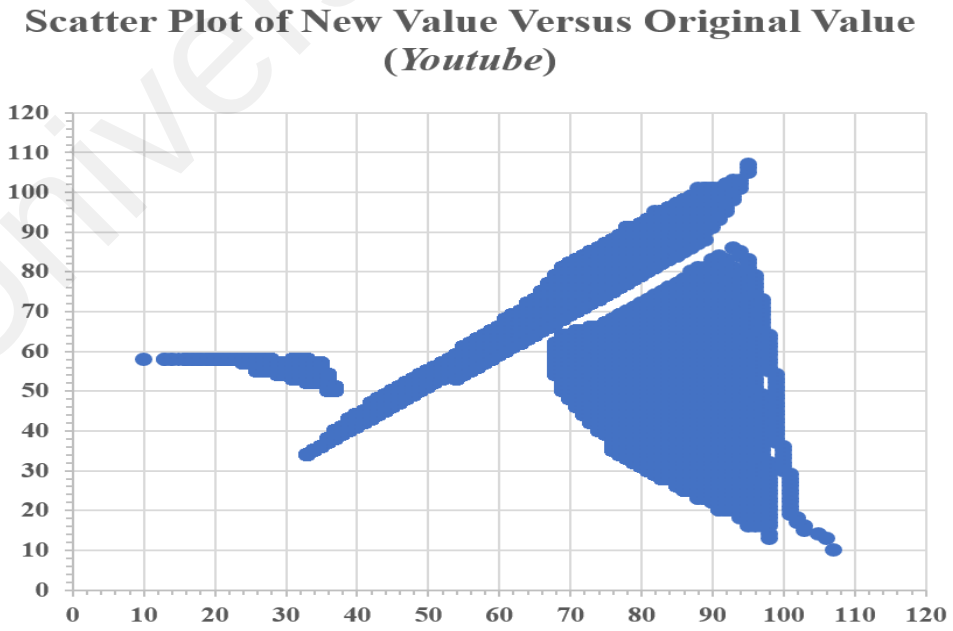

Figure 5.3: Scatter plots of new value versus original value in *Youtube* (*MinSwap*).

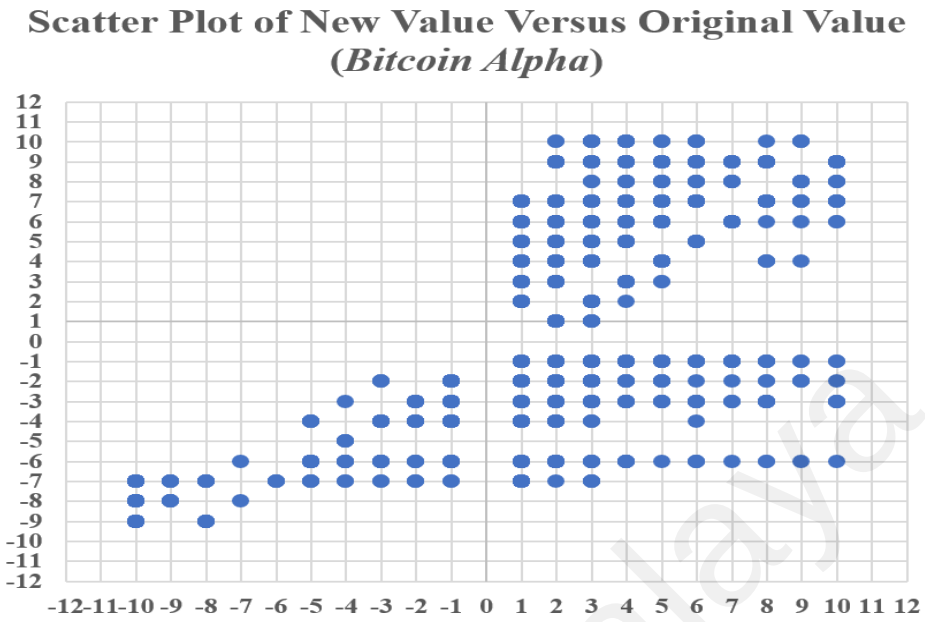

Figure 5.4: Scatter plots of new value versus original value in *Bitcoin Alpha* (*δ-MinSwapX*).

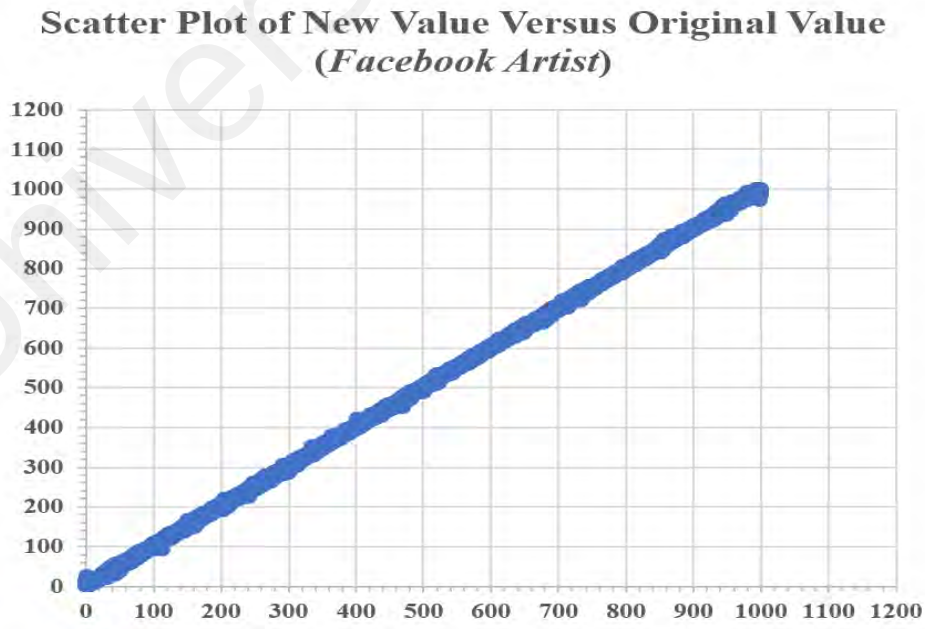

Figure 5.5: Scatter plots of new value versus original value in *Facebook Artist* (*δ-MinSwapX*).

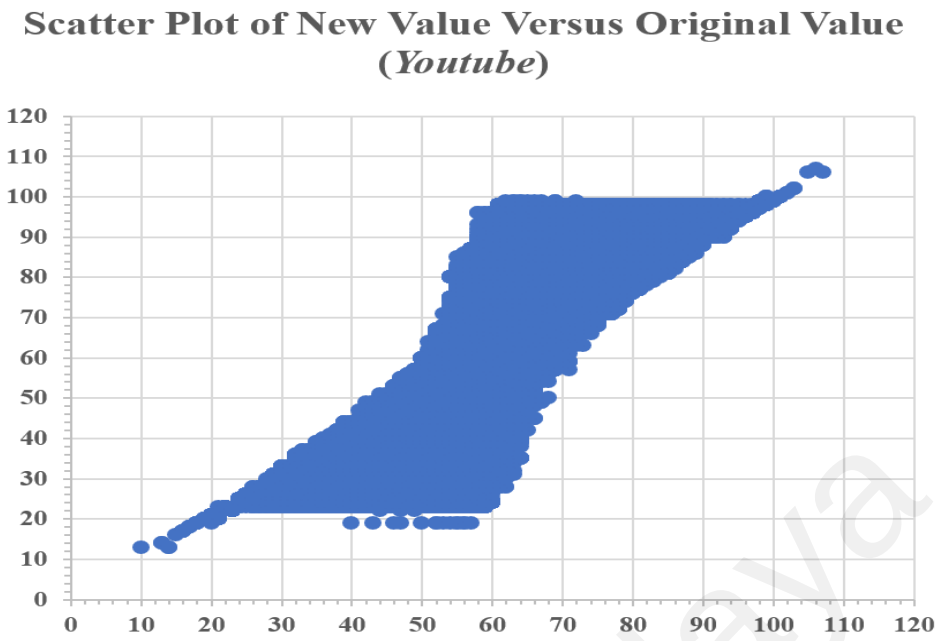

Figure 5.6: Scatter plots of new value versus original value in *Youtube* (*δ-MinSwapX*).

Table 5.2: Comparison of privacy protection.

Privacy Preservation	Edge Weight Disclosure		Link Disclosure	Identity Disclosure	
	A	U		A	U
S. Das et al. (2010a); Li et al. (2017); L. Liu et al. (2010, 2008, 2009); Wang, Tsai, et al. (2013); Wang et al. (2011)	✔	✖	✖	✖	✖
Cormode et al. (2008); Fard and Wang (2015); Ying and Wu (2008); Zheleva and Getoor (2007)	✖	✖	✔	✖	✖
Campan and Truta (2008); Day et al. (2016); Hay et al. (2008); K. Liu and Terzi (2008); Macwan and Patel (2018); Tai et al. (2011); Zhou and Pei (2008, 2011); Zou et al. (2009)	✖	✖	✖	✔	✖
Cheng et al. (2010); Hay et al. (2009, 2007); P. Liu et al. (2017, 2019); Nissim et al. (2007); L. Peng et al. (2017); Rastogi et al. (2009); Ying et al. (2009); Zhan et al. (2017)	✖	✖	✔	✔	✖
Babu et al. (2013); Chen and Zhu (2015); Q. Liu et al. (2016); Skarkala et al. (2012); Wang, Shih, et al. (2013); Yuan and Chen (2011)	✔	✖	✖	✔	✖
MinSwap	✔	✔	✖	✔*	✖
δ-MinSwapX	✔	✔	✔	✔	✔

A is anonymity, U is unlinkability and ✔ indicates partially addressed.

In this thesis, we proposed two schemes that address edge weight disclosure, link disclosure and identity disclosure based on unlinkability notion, which has not been considered in prior work. We compare our work with some related literature discussed in chapter 2 in terms of the privacy protections and summarize the comparisons in Table 5.2. We further analyze the privacy level rendered by our work in proposition 3, 4, 8, 9 and 10.

In the previous work as shown in Table 5.2, anonymity of edge weight is achieved through the process of data perturbation, k-anonymization, differential privacy and generalization, such that an edge weight could not be reidentified with high probability. As shown in our gap analysis, these schemes do not provide unlinkability feature to the edge weight data. That is, the original edge weight value could be inferred from its perturbed value. In contrast, our schemes provide anonymity and unlinkability ($w \neq w'$) simultaneously, such that there does not exist an injective mapping between w and w' (one value can be assigned to different values in the published data). The edge weight protection rendered in our schemes is higher than k-anonymization schemes as the distinct values in network data are diverse ($n > k$), as shown in proposition 8. All the edge weights are modified in *MinSwap*, providing a certain amount of node protection to the user. However, the scheme is vulnerable to structural attacks as no structural modification is applied.

δ-MinSwapX is proposed to provide additional link and node protection. Randomization is deployed to randomly modify the structural information according to the edge betweenness. Random fake edge addition hides the presence of true link in the published graph, and thus prevents the link disclosure, regardless of the background knowledge an adversary may possess, as shown in proposition 9. Furthermore, fake node addition hides the true nodes in the published data. The number of fake nodes and fake edges added by our schemes depend on the original data itself, thus the number of fake nodes and edges cannot be inferred by an adversary with high confidence level. Since the edges are

randomized, the change of structural information is randomized. An adversary cannot simply map the auxiliary structural information to attack the published data in order to infer the link and identity of a user. In addition, *node unlinkability* further guarantees that the edge weight information cannot be linked to its corresponding user in the published data.

Proposition 8. The probability of edge weight disclosure, $P(w_a) = \frac{1}{N-1}$ for *MinSwap* and $\frac{1}{N-|\boldsymbol{W'(a \cup b)}|}$ for δ-*MinSwapX*, where N = number of distinct edge weight values in $\boldsymbol{W}$.

Proof. For an edge weight value w, every other edge weight values in the original data has equal chance of being the new edge weight w' of a victim a. Hence, the probability of edge weight disclosure of victim a under *MinSwap*, $P(w_a) = \frac{1}{N-1}$. If an adversary has a high confidence level, ϵ that the true edge weight lies in a set of x values ($x \leq N-1$), then $P(w_a) = \epsilon\frac{1}{x} + (1-\epsilon)\frac{1}{N-1-x}$. An adversary may not have high confidence level regarding the exact original edge weight values. Hence, when x approaches $N-1$, ϵ approaches to 1, and $P(w_a)$ approaches $\frac{1}{N-1}$.

In the case of δ-*MinSwapX*, an adversary learns that the true edge weight $\in [\boldsymbol{W'(a \cup b)}]^c$. Hence, $P(w_a) = \frac{1}{N-|\boldsymbol{W'(a \cup b)}|}$. $P(w_a)$ is arbitrary small since N is arbitrarily large in scalable social network data as shown in subsection 5.1. This completes the proof. □

Proposition 9. The probability of inferring the presence of link under δ-*MinSwapX* $= 1-\delta$ and the probability of reidentification of the true link $= \frac{(1-\delta)m}{(1-\delta)m+m_{Add}}$, where m_{Add} is the number of fake edges added.

Proof. The probability of inferring the presence of link $= 1-\delta$, as δ of the original link are removed from the graph under δ-*MinSwapX*.

The probability of link reidentification = fraction of true link in the published data = $\frac{(1-\delta)m}{(1-\delta)m+m_{Add}}$. This is the same privacy level rendered in (Fard & Wang, 2015). This completes the proof. □

Proposition 10. The probability of identity disclosure of node a under δ-*MinSwapX* = $max\ [\frac{1}{n+n_{Add}}, \prod_{i=1}^{D_a} \frac{1}{N-|\boldsymbol{W'(a\cup b)}|}, \frac{[\sigma(n_2-n_1)+n_1]}{n_1n_2n_3}]$, where n_1 is the number of edges deleted for node a, n_2 is the number of edges added for node a, n_3 is the number of node with D_a in the published data and σ is the confidence level that a node undergoes edge deletion.

Proof. There are three possible alternatives to reidentify a victim a using edge weight and structural data as background knowledge:

a) Brute-force: Every node in the published data has equal chance of being victim a. Hence, the probability of reidentification of victim a, $P(a) = \frac{1}{n+n_{Add}}$.

b) Reconstruct the original edge weight from the published graph and deploy linkage attack: The probability of inferring all the true edge weights is, *P(All edge weights are true)* = $\prod_{i=1}^{D_a} \frac{1}{N-|\boldsymbol{W'(a\cup b)}|}$. By matching the auxiliary edge weight values with the reconstructed edge weights, in the worst case, there is an exact match where $P(a) = \prod_{i=1}^{D_a} \frac{1}{N-|\boldsymbol{W'(a\cup b)}|}]$.

c) Reconstruct the original structural graph from the published data and deploy linkage attack: Every node is subjected to either edge deletion or edge addition. The change of degree of node a is $[-n_1, 0)$ or $(0, n_2]$. Given an adversary has a confidence level of σ that a node undergoes edge deletion, then the probability of inferring the correct degree, $P(D_a) = \frac{\sigma}{n_1} + \frac{1-\sigma}{n_2}$. If there are n_3 nodes with D_a in the published data, $P(a) = \frac{\sigma(n_2-n_1)+n_1}{n_1n_2n_3}$.

This completes the proof. □

5.3 Efficiency Evaluation

Figure 5.7 demonstrates the running time of both *MinSwap* and *δ-MinSwapX* for δ = 0, 0.2, 0.4, 0.6, 0.8 and 1. At δ = 0, only edge weight modification is applied on the data. At δ = 1, the time taken is zero as all edges are removed from the data set and a null graph is obtained. Hence, 1-*MinSwapX* is not considered in the later evaluations. Generally, *δ-MinSwapX* consumes more time compared to *MinSwap*. The structural modification (Algorithm 4.4) of *δ-MinSwapX* constitutes to a high running time, especially when the data sets are scalable as observed in ***Facebook Artist*** and ***Youtube***. However, both schemes are efficient when the graph is relatively small as shown in ***Bitcoin Alpha*** (< 52.54s).

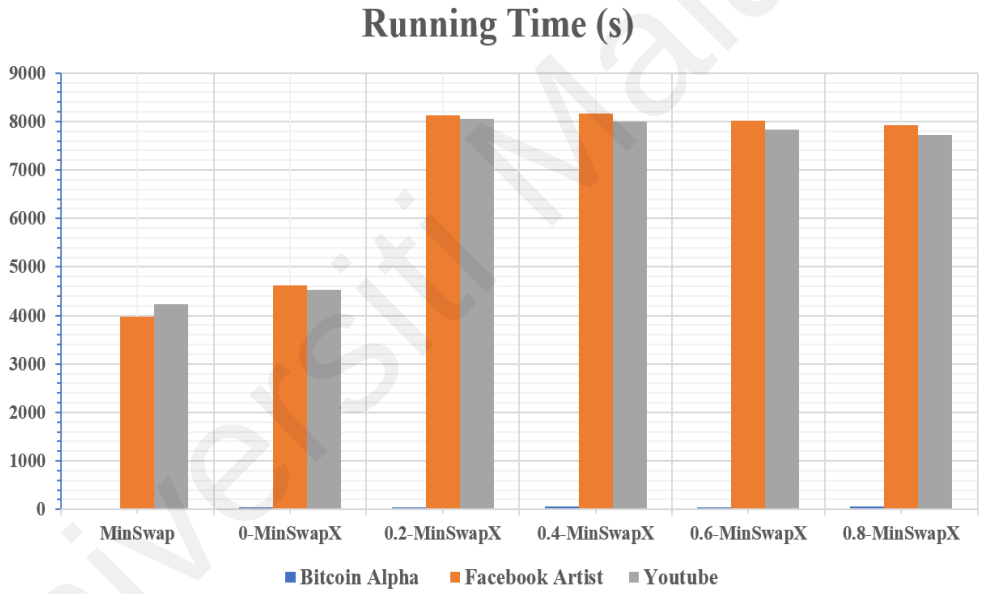

Figure 5.7: Running time (s) according to data sets.

Table 5.3: Running time (s) according to data sets.

Data Set	*MinSwap*	*δ-MinSwapX*					
		0	0.2	0.4	0.6	0.8	1
Bitcoin Alpha	32.93	40.85	50.62	51.92	51.13	52.54	0
Facebook Artist	3983.12	4623.21	8134.69	8163.08	8020.33	7925.12	0
Youtube	4235.69	4532.12	8052.32	8000.32	7832.45	7723.23	0

As shown in Algorithm 4.1, there is a loop of m number of edge weight data during possible set determination and candidate selection. Hence, the time complexity of *MinSwap* is $O(m)$. In the case of δ-*MinSwapX*, there is a loop of m number of edge weight data in Algorithm 4.3. Furthermore, there are two loops of m_{Add} and n_{Add} number of edge weight data during fake edge addition and edge weight addition in Algorithm 4.4. Hence, the total time complexity of δ-*MinSwapX* = $O(m) + O(m_{Add}) + O(n_{Add}) = O(m)$ in the worst case. The linear complexity of both schemes implies **the feasibility of our schemes in anonymizing scalable data**. As shown in Figure 5.7, the running time increases linearly with the data size.

5.4 Utility Evaluation

We analyze several statistical metrics to verify the statistical properties preservation strength as one of the unique feature rendered by our work. In addition, we study a set of common graph metrics and shortest path analysis, which were similarly adopted in (Cheng et al., 2010; Hay et al., 2007; L. Liu et al., 2009; Wang, Tsai, et al., 2013) to validate the utility of the anonymized graph.

5.4.1 Statistical Properties Analysis

We evaluate the impact of *MinSwap* and δ-*MinSwapX* on the statistical properties of each data set. Figure 5.8, 5.9 and 5.10 show the edge weight distribution of each data set post implementation of *MinSwap* and δ-*MinSwapX*. Kolmogorov-Smirnovb test at confidence level = 0.05 is utilized to verify the distribution preservation.

Based on Figure 5.8, 5.9 and 5.10, we observed that the data distribution of all three data sets are fully preserved at rate = 100% under *MinSwap*. Meanwhile, for the case of δ-*MinSwapX*, the degree of change of the data distribution increases as the value of δ increases. When the value of δ reaches 0.8, the total frequency of edge weight data

decreases and the data distributions are observed to be approaching a uniform distribution. In other words, the frequency of each edge weight is observed to be almost equal likely the same, especially in Figure 5.9 and 5.10.

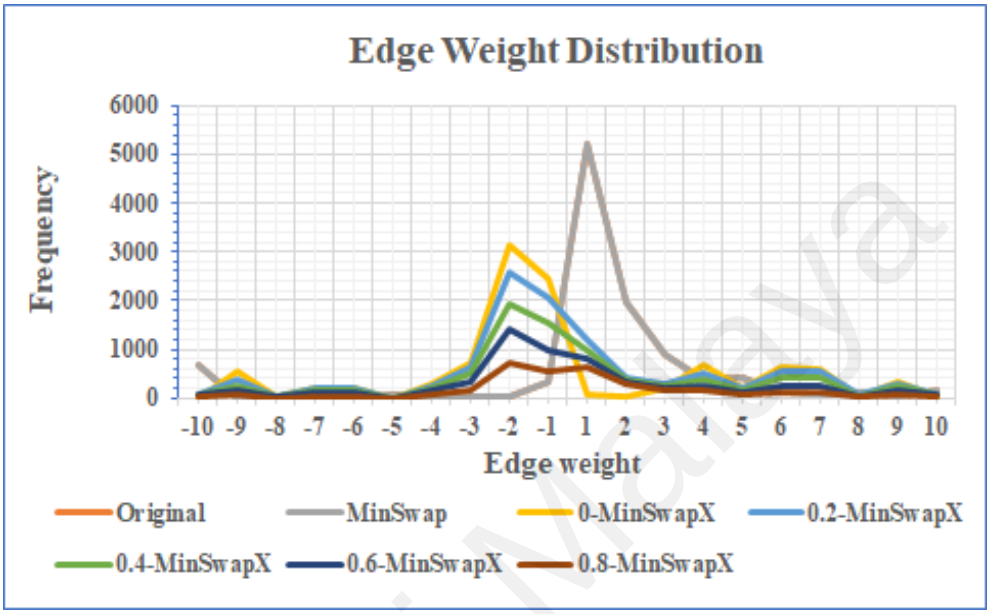

Figure 5.8: Edge weight distribution of *Bitcoin Alpha*.

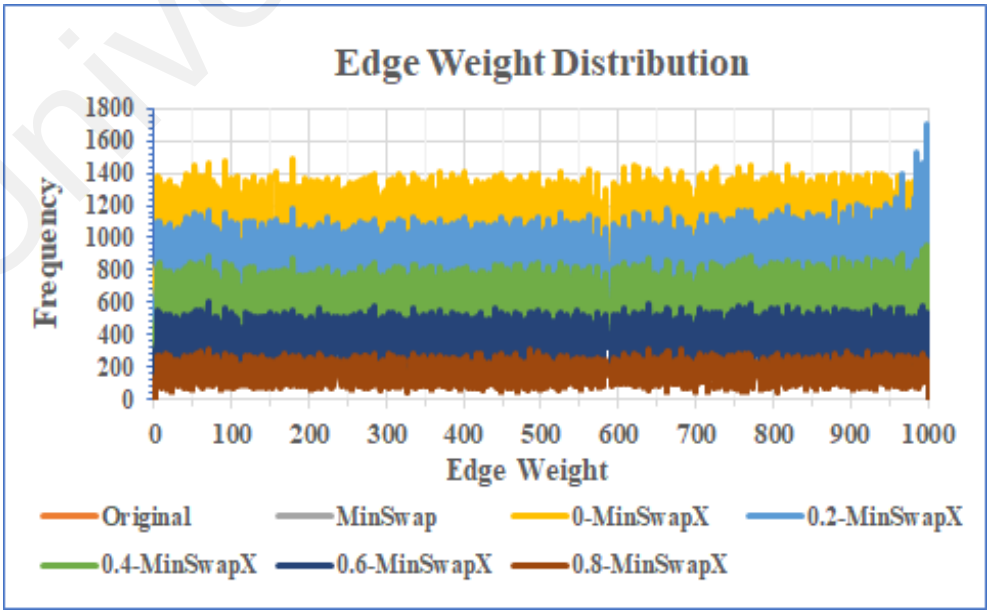

Figure 5.9: Edge weight distribution of *Facebook Artist*.

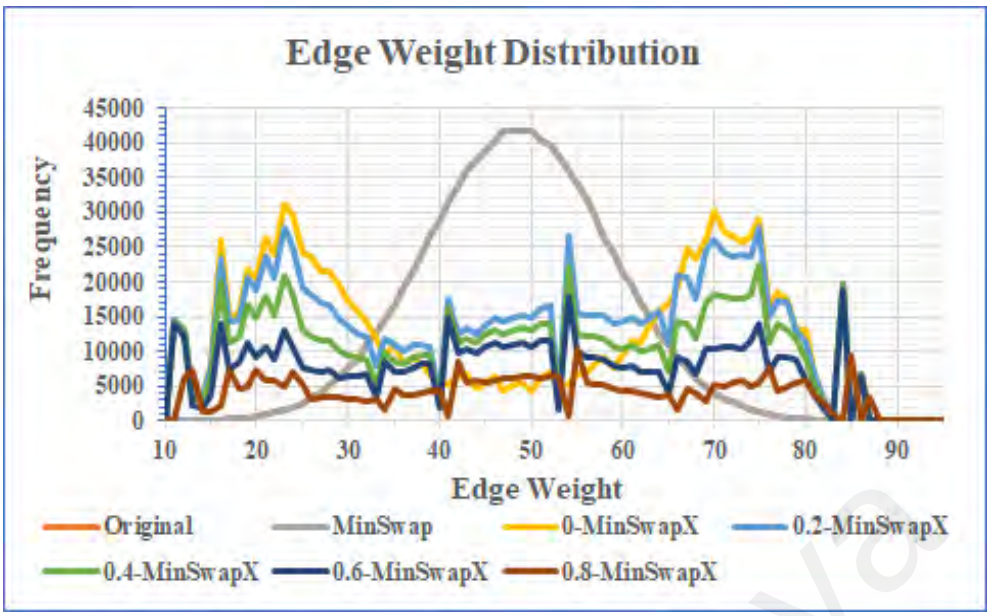

Figure 5.10: Edge weight distribution of *Youtube*.

Table 5.4, 5.5 and 5.6 show the detailed experimental results of *MinSwap* and *δ-MinSwapX* on ***Bitcoin Alpha***. The statistical properties of all data sets are preserved at 100% rate under *MinSwap* as the edge weights are inter-swapped from the original data without addition or deletion of edge weight. The preservation of statistical properties is one of the unique features rendered by *MinSwap* compared to other works. *δ-MinSwapX* is not designed to preserve the statistical properties. However, it provides well preservation on the mean and standard deviation of data.

Table 5.4: Statistical properties analysis of *Bitcoin Alpha*.

Bitcoin Alpha	Original Data	*MinSwap*	*δ-MinSwapX*				
			0	0.2	0.4	0.6	0.8
Mean	1.05	1.05	-0.35	-0.03	0.01	0.05	0.25
Median	1.00	1.00	-1.00	-1.00	-1.00	-1.00	-1.00
Mode	1.00	1.00	-2.00	-2.00	-2.00	-2.00	-2.00
Standard Deviation	3.50	3.50	4.32	4.14	4.14	4.06	3.81
Sample Variance	12.22	12.22	18.62	17.17	17.15	16.52	14.49
Kurtosis	4.63	4.63	-0.14	0.10	0.12	0.26	0.69
Skewness	-1.62	-1.62	0.39	0.23	0.23	0.19	0.08
Range	20.00	20.00	19.00	20.00	20.00	20.00	20.00
Minimum	-10.00	-10.00	-9.00	-10.00	-10.00	-10.00	-10.00
Maximum	10.00	10.00	10.00	10.00	10.00	10.00	10.00
Sum	11128.00	11128.00	-3716.00	-271.00	63.00	283.00	868.00
Count	10554.00	10554.00	10554.00	10612.00	8169.00	5769.00	3417.00

Table 5.5: Statistical properties analysis of *Facebook Artist*.

Facebook Artist	Original Data	*MinSwap*	*δ-MinSwapX*				
			0	0.2	0.4	0.6	0.8
Mean	500.53	500.53	500.53	510.70	505.78	504.11	503.49
Standard Error	0.32	0.32	0.32	0.35	0.41	0.50	0.70
Median	501.00	501.00	500.00	514.00	508.00	507.00	505.00
Mode	838.00	838.00	179.00	999.00	998.00	69.00	681.00
Standard Deviation	288.82	288.82	288.81	292.21	290.13	289.27	289.30
Sample Variance	83415.84	83415.84	83411.34	85386.50	84177.41	83674.41	83694.64
Kurtosis	-1.20	-1.20	-1.20	-1.21	-1.21	-1.20	-1.20
Skewness	0.00	0.00	0.00	-0.03	-0.02	-0.02	-0.01
Range	999	999	999	999	999	999	999
Minimum	1	1	1	1	1	1	1
Maximum	1000	1000	1000	1000	1000	1000	1000
Sum	410087357	410087357	410086576	346352883	254742821	169339690	85417899
Count	819306	819306	819306	678190	503664	335916	169653

Table 5.6: Statistical properties analysis of *Youtube*.

Youtube	Original Data	*MinSwap*	*δ-MinSwapX*				
			0	0.2	0.4	0.6	0.8
Mean	59.52	59.52	59.70	60.35	60.29	60.05	60.02
Median	60.00	60.00	60.00	62.00	61.00	61.00	60.00
Mode	58.00	58.00	35.00	35.00	87.00	96.00	66.00
Standard Deviation	10.00	10.00	23.18	22.11	21.53	21.24	21.31
Sample Variance	100.00	100.00	537.12	488.67	463.34	451.03	454.17
Kurtosis	0.00	0.00	-1.59	-1.41	-1.27	-1.14	-1.13
Skewness	-0.01	-0.01	-0.01	-0.08	-0.08	-0.05	-0.04
Range	97	97	94	93	93	93	93
Minimum	10	10	13	13	13	13	13
Maximum	107	107	107	106	106	106	106
Sum	62407638	62407638	62597938	63280373	54040360	38842178	20941767
Count	1048571	1048571	1048571	1048575	896335	646838	348913

Table 5.7: Changes of statistical properties after *MinSwap* and *δ-MinSwapX*.

Data Set	*MinSwap*			*δ-MinSwapX*		
	Dis	Mean	SD	Dis	Mean	SD
Bitcoin Alpha	✔	✔	✔	✖	✖	✔
Facebook Artist	✔	✔	✔	✖	✔	✔
Youtube	✔	✔	✔	✖	✔	✔

*Dis is distribution and SD is standard deviation.

We summarize the changes of statistical properties after *MinSwap* and *δ-MinSwapX* in Table 5.7, with emphasis on the distribution, mean and standard deviation as some important parameters of a statistical database.

5.4.2 Shortest Path Analysis

The Dijkstra algorithm (Deng et al., 2012) is used to determine the shortest paths between all reachable node pairs and evaluate the corresponding shortest path length. We consider the change of shortest path length of the most influential nodes as it is infeasible to evaluate the shortest paths of all reachable nodes in scalable networks. Figure 5.11

shows the change of average shortest path length.

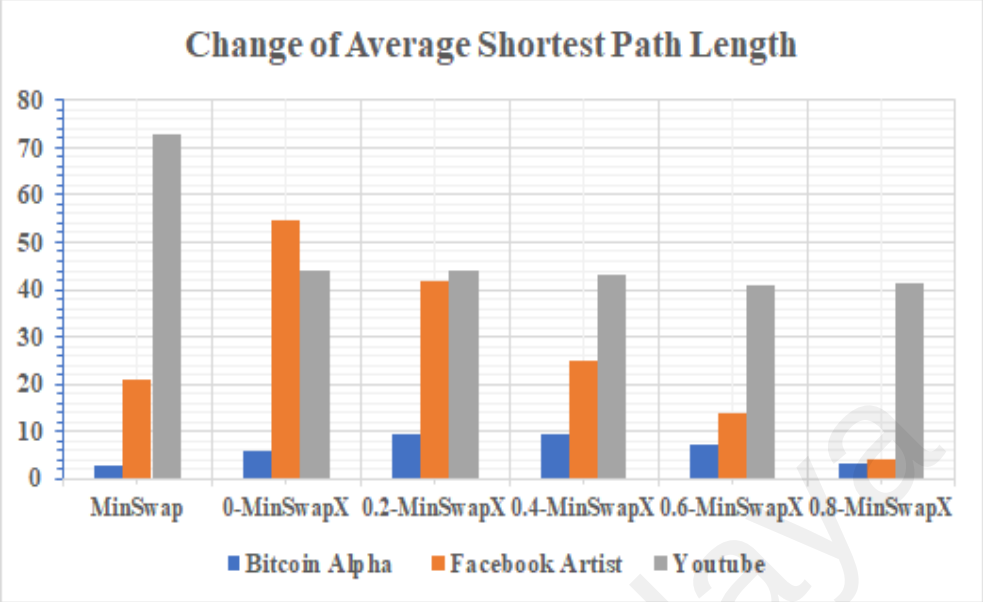

Figure 5.11: Change of average shortest path length.

Table 5.8: Change of average shortest path length.

Data Set	*MinSwap*	*δ-MinSwapX*				
		0	0.2	0.4	0.6	0.8
Bitcoin Alpha	2.5414	5.8107	9.3962	9.4528	7.2092	3.1954
Facebook Artist	20.7854	54.6460	41.7887	24.8851	13.9443	4.2006
Youtube	72.8543	43.8717	44.1327	43.0388	40.6952	41.1398

All data sets show low change of average shortest path length compared to the range of the edge weight values, except for ***Youtube***. ***Youtube*** is normally distributed data. The edge weights near the mean may inter-swapped between each other. This leads to the scarcity of possible candidates for the data at the two tails of the distribution. Larger noises may be injected to data at the two tails and this results in a larger change of average shortest path length in ***Youtube***.

5.4.3 Network Centrality Analysis

We examine the ratio of fake node and fake edge added to the network as one of the utility metrics. We examine some common graph metrics to evaluate the information loss in δ-*MinSwapX*. *MinSwap* preserves the network structure as no structural modification is applied, and thus is omitted. Clustering coefficient is a measure of the extent to which nodes in a graph tend to cluster together. Closeness is the inverse of average shortest path length. Normalized connectivity centralization measures the degree to which a graph resembles a star graph[4] topologically.

As shown in Figure 5.12 and 5.13, the ratio of fake node and fake edge decrease as the δ increases. Note that all the original nodes are preserved in the published data. Regardless of the value of δ, all the structural data of a node are modified. The higher the value of δ, the larger the amount of modification.

As the value of δ increases, the edge deletion process compensates the effect of edge addition, which eventually modifies the original graph into a null graph. Therefore, the metrics decreases as shown in Figure 5.14, 5.15, 5.16 and 5.17. The preservation of graph centrality is relatively high and the metrics change steadily over δ.

[4] A star graph is a tree on n nodes with one node having degree n-1 and the other n-1 nodes having degree 1

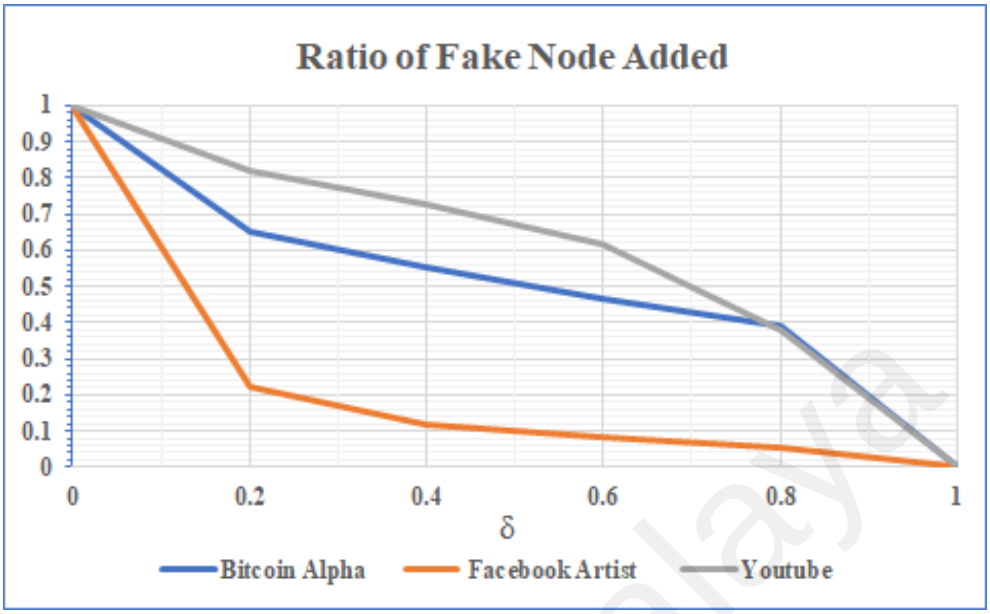

Figure 5.12: Ratio of fake node added.

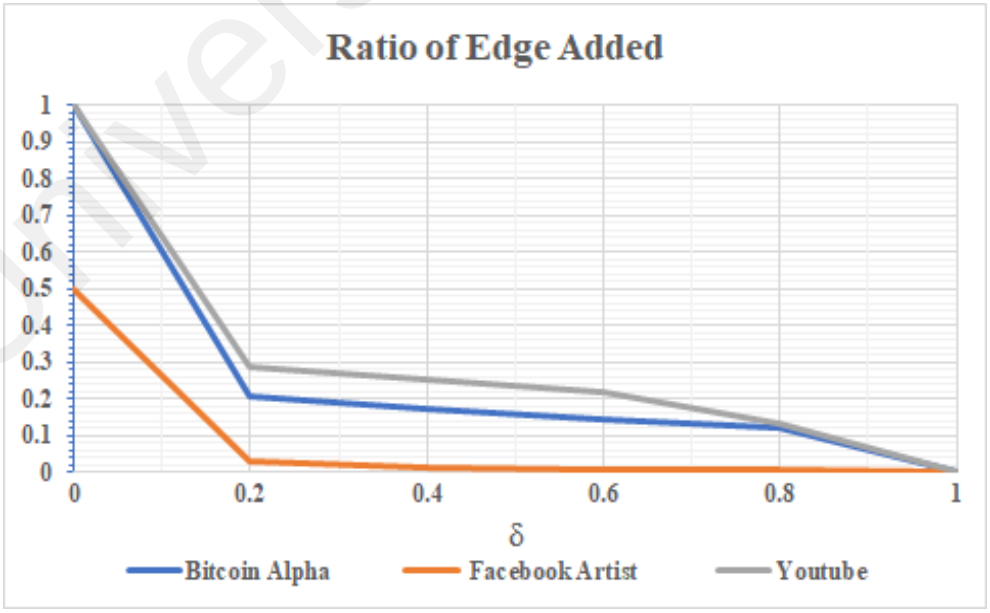

Figure 5.13: Ratio of fake edge added.

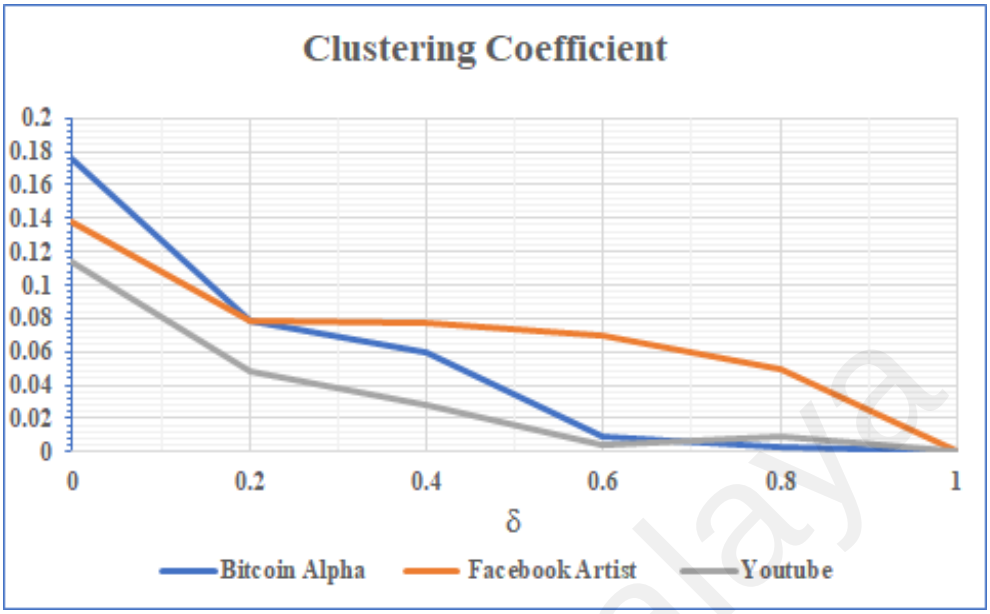

Figure 5.14: Clustering coefficient.

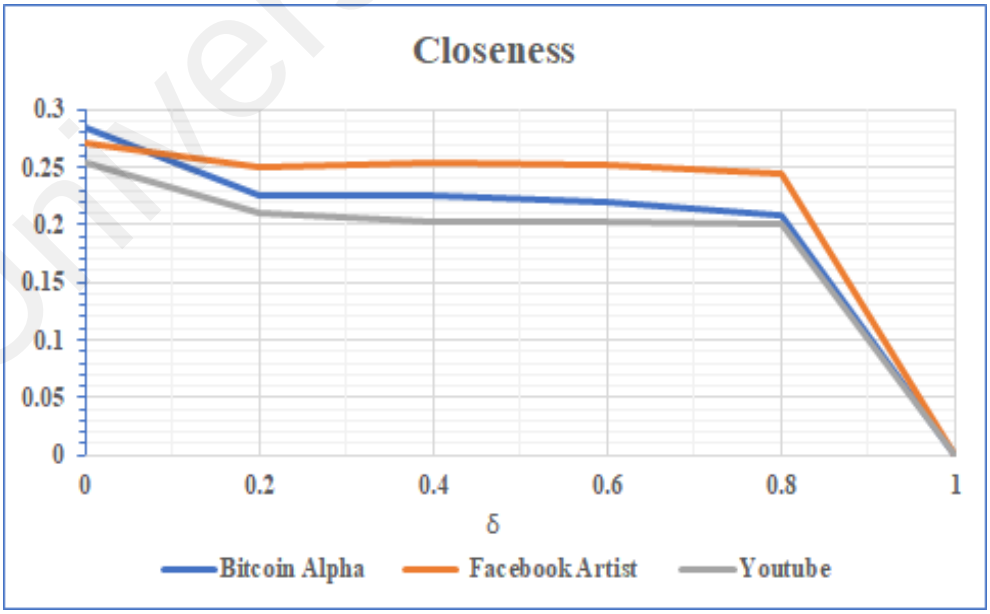

Figure 5.15: Closeness.

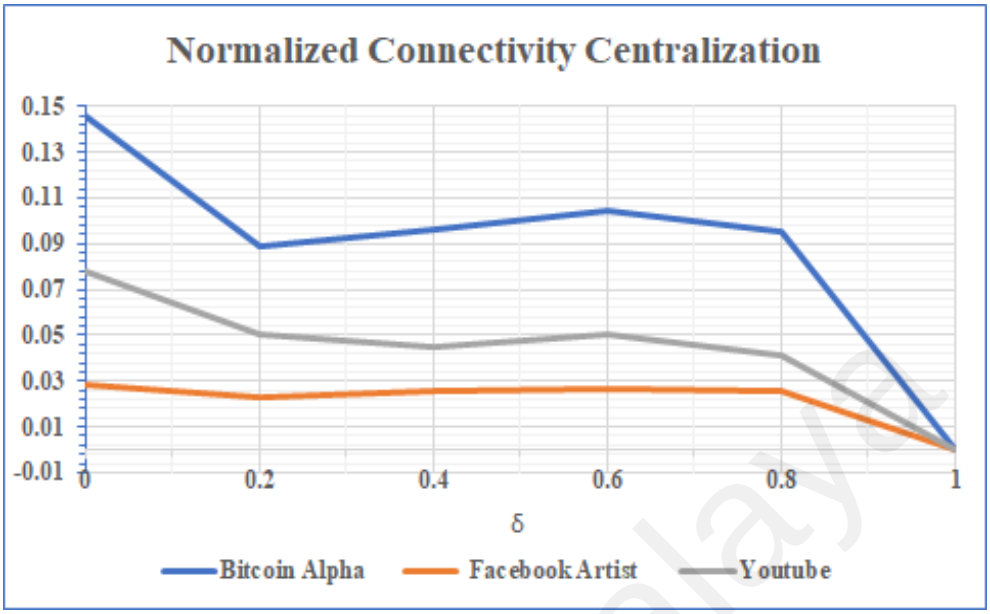

Figure 5.16: Normalized connectivity centralization.

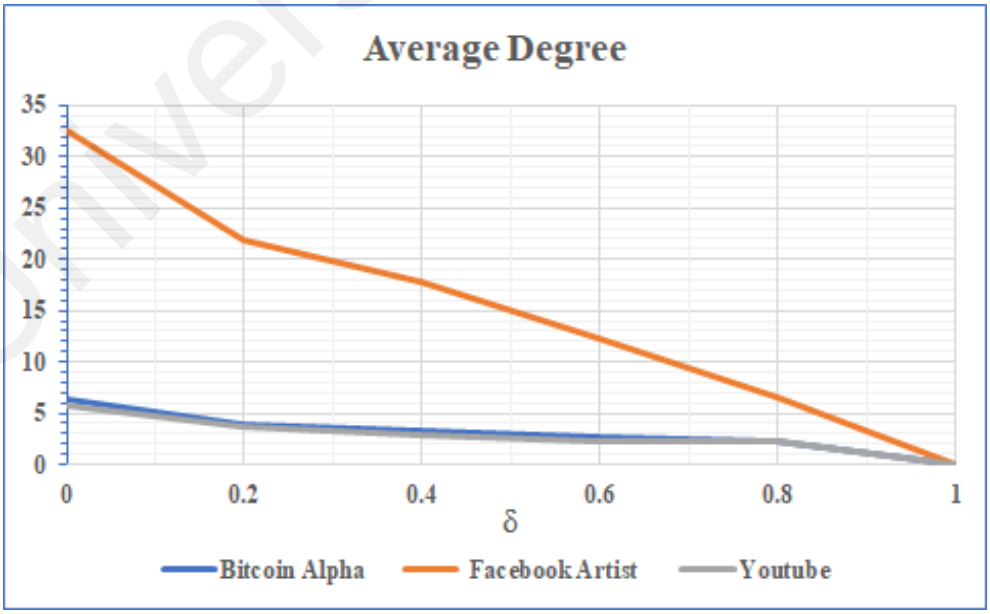

Figure 5.17: Average degree.

Tables 5.9, 5.10, 5.11, 5.12, 5.13 and 5.14 shows the experimental results on network centrality analysis.

Table 5.9: Ratio of fake node added.

Data Set	*δ-MinSwapX*					
	0	0.2	0.4	0.6	0.8	1
Bitcoin Alpha	1	0.6536	0.5533	0.4659	0.3933	0
Facebook Artist	1	0.2251	0.1195	0.0811	0.0573	0
Youtube	1	0.8202	0.7249	0.6170	0.3776	0

Table 5.10: Ratio of fake edge added.

Data Set	*δ-MinSwapX*					
	0	0.2	0.4	0.6	0.8	1
Bitcoin Alpha	1	0.2056	0.1741	0.1465	0.1237	0
Facebook Artist	0.5	0.0278	0.0147	0.0100	0.0071	0
Youtube	1	0.2883	0.2548	0.2169	0.1328	0

Table 5.11: Clustering coefficient.

Data Set	*δ-MinSwapX*					
	0	0.2	0.4	0.6	0.8	1
Bitcoin Alpha	0.176	0.078	0.06	0.009	0.003	0
Facebook Artist	0.138	0.079	0.077	0.07	0.05	0
Youtube	0.114	0.048	0.028	0.004	0.009	0

Table 5.12: Closeness.

Data Set	*δ-MinSwapX*					
	0	0.2	0.4	0.6	0.8	1
Bitcoin Alpha	0.2846	0.2251	0.2247	0.2189	0.2079	0.0000
Facebook Artist	0.2713	0.2504	0.2535	0.2530	0.2445	0.0000
Youtube	0.2531	0.2110	0.2032	0.2022	0.2009	0.0000

Table 5.13: Normalized connectivity centralization.

Data Set	*δ-MinSwapX*					
	0	0.2	0.4	0.6	0.8	1
Bitcoin Alpha	0.146	0.089	0.096	0.104	0.095	0
Facebook Artist	0.028	0.023	0.026	0.027	0.026	0
Youtube	0.078	0.05	0.045	0.05	0.041	0

Table 5.14: Average degree.

Data Set	*δ-MinSwapX*					
	0	0.2	0.4	0.6	0.8	1
Bitcoin Alpha	6.358	3.876	3.2	2.572	2.238	0
Facebook Artist	32.43	21.917	17.811	12.302	6.48	0
Youtube	5.69	3.627	2.826	2.301	2.294	0

5.5 Summary

In this chapter, we evaluated the strengths of the proposed work theoretically and empirically. We have shown that our work provide higher privacy level to the users, in terms of edge weight, link and identity protection compared to other existing work, by rendering unlinkability in a social network. From the utility aspect, *MinSwap* preserves the statistical properties of edge weight data at rate = 100%. Regardless of the value of the δ, all the structural information of each node are changed, providing a considerable amount of protections to the user. Furthermore, the performance of shortest path analysis is substantial and the network properties are well preserved, considering the degree of privacy protection provided.

CHAPTER 6: CONCLUSION

In this chapter, we highlight our key contributions and discuss the directions for future work in the topics addressed in this thesis.

6.1 Concluding Remarks and Summary of Contributions

This thesis studied the problems of preserving privacy in social networks. Releasing network data to a third party is a common practice of service providers who have some specific interests in certain analysis and data mining outcomes of their data. Publication of raw data prompts to unintended privacy disclosures, such as identity disclosure, link disclosure and edge weight disclosure when sensitive information of a user is disclosed in the published data to an adversary. To address these privacy issues, anonymization is performed as a standard process to modify an original data into an anonymized data that satisfy some privacy constraints. However, data anonymization may lead to excessive information loss, for instance, in the context of statistical properties and network topological properties. This may renders the published data to be useless in some applications. Simultaneous preservation of both data privacy and utility remains a challenging topic and a more efficient and effective network data publication scheme is on demand.

An extensive literature review was presented to evaluate the level of privacy and utility preservation in related anonymization schemes. Our gap analysis has shown that prior work were built on the anonymity notion, in which previous schemes lack an additional unlinkability notion to further protect a user. Unlinkability requires that there is no one-to-one mapping between the original and published data. Hence, no information can be inferred from the published data with high confidence level regarding a user, regardless of the background knowledge possessed by an adversary. Furthermore, prior schemes incur excessive distortion to the original data, in terms of its statistical properties, graph

structure and shortest path length. This affects the published data to show deviated results from the original data in most analysis. These privacy and utility limitations are our focus and have been addressed in this research.

In this thesis, we constructed two new anonymization schemes to guarantee a secure and useful sharing of social network data. Particularly, the contributions of this thesis are as follows:

1. This thesis proposed two novel privacy models to provide additional layers of privacy protection to a network user, namely *edge weight unlinkability* and *node unlinkability*. *Edge weight unlinkability* guarantees that all sensitive edge weight values of a user are perturbed, such that the published values cannot be reverse-engineered to infer the original values. *Node unlinkability* provides a stronger layer of privacy on top of *edge weight unlinkability*. It guarantees that the associations between the edge weights and the nodes are broken, such that no auxiliary edge weight information could be utilized to infer the identity of a user in the published data. To the best of our knowledge, our work are the **first privacy model in weighted social networks that are built on the unlinkability notion.**
2. This thesis deployed the formulated notions as a framework to design two new anonymization schemes, namely ***MinSwap* that addresses edge weight disclosure** and ***δ-MinSwapX* that addresses identity disclosure, link disclosure and edge weight disclosure** simultaneously. Perturbation is deployed in *MinSwap* based on the idea of data swapping to guarantee *edge weight unlinkability*, while fully preserve the overall statistical properties of edge weight data. Moreover, perturbation is applied in *δ-MinSwapX* to minimally perturb the edge weight data to provide an extra layer of *node unlinkability* to a user. Randomization is also deployed in *δ-MinSwapX* to modify the structural data to protect a user against identity disclosure and link

disclosure. Since the links are randomized according to the edge betweenness, this allows greater preservation of important links in the original data, while prevents the linking of structural background knowledge to a user in the published data.

3. This thesis presented an extensive analysis on the strength of the proposed notions and schemes in terms of privacy, efficiency and utility. We performed simulations using scalable network data and demonstrated that our schemes are efficient and usable for real world implementation. **Comparisons with other relevant schemes showed that our work maintain a high data privacy and utility simultaneously.** To be precise, our schemes preserve the statistical properties of edge weight data and the network topological properties such as network centrality. Furthermore, our schemes achieve high preservation rate of average shortest path length. These utility improvements render the published data to show more accurate results in analysis.

6.2 Directions for Future Work

This section discusses a number of directions for future research in line with the problems studied in this thesis. Some possible extensions are presented as follows.

1. **Standardization of privacy model**

   Server providers collecting data from users are required to comply with a number of privacy policies to protect the privacy of a user. This may require the server providers to install systems and processes in place to maintain compliance. However, there is no clear indication of which privacy model and protection level should be adopted. Privacy issues might arise when there are different variations of anonymized data given an original database. This is possible as an entity may have multiple profiles in different social networks and the sensitive personal information contained in these profiles are similar. Although the data published by each publisher satisfy certain

privacy requirements, the released data could be variant from each other due to the difference in the implementation of privacy models and privacy parameters setting. An adversary could still intrude the data privacy by combining both published data together. It may be of interest to develop a standardization of privacy protection for privacy policy compliance as one of the subjects of future research.

2. **Privacy preserving network data publication in a distributed and dynamic environment.**

   This thesis considers data publication in a centralized and static environment, such that only one service provider publishes the data to third party data recipients at a time interval. We may consider the implementation of the proposed schemes in a distributed environment, where there are multiple service providers who publish their data independently to a data pool with the possibility of data overlapping. The problem is on how to anonymize and analyze the aggregated data that consists of anonymized data from each publisher. Furthermore, data are collected and published continuously in a dynamic network. The information contained in profiles could be updated from time to time and required to be reflected in the anonymized data. It is of interest to design a privacy model that considers data publication in a distributed and dynamic environment.

3. **Personalized privacy protection.**

   Different users have different preferences regarding their privacy level. Some users prefer high privacy preservation, such that none of their data should be published. At the same time, there are users who are neutral in their privacy preference, such that their data could be published selectively according to the privacy laws and policies. Different level of privacy protection could be applied to preserve a greater extent of data utility. However, this may lead to other privacy issues. Given two

connected entities, node a with high privacy preference and node b with no privacy preference, the data publication of node b may intrude the privacy of node a. For example, the disclosure of information that node b is connected to node a may imply the link of node a. In a social network where people tend to have friends who are also friends with each other. Such information may provide additional background knowledge to an adversary in inferring the true neighborhood graph of node a. One of the possible issues that arises is on how to maintain the privacy protection to both individuals while allowing the publication of such information. To address this problem, further study on personalized privacy model is required.

REFERENCES

Al-Garadi, M. A., Varathan, K. D., Ravana, S. D., Ahmed, E., Mujtaba, G., Khan, M. U. S., & Khan, S. U. (2018). Analysis of online social network connections for identification of influential users: Survey and open research issues. *ACM Computing Surveys (CSUR)*, *51*(1), 1–37.

Arruda, A., Friendship, R., Carpenter, J., Hand, K., & Poljak, Z. (2016). Network, cluster and risk factor analyses for porcine reproductive and respiratory syndrome using data from swine sites participating in a disease control program. *Preventive Veterinary Medicine*, *128*, 41–50.

Atzmueller, M., Hanika, T., Stumme, G., Schaller, R., & Ludwig, B. (2016). Social event network analysis: Structure, preferences, and reality. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, 613–620.

Babu, K. S., Jena, S. K., Hota, J., & Moharana, B. (2013). Anonymizing social networks: A generalization approach. *Computers & Electrical Engineering*, *39*(7), 1947–1961.

Bensimessaoud, S., Badache, N., Benmeziane, S., & Djellalbia, A. (2016). An enhanced approach to preserving privacy in social network data publishing. *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*, 80–85.

Berent, M. K., Krosnick, J. A., & Lupia, A. (2016). Measuring voter registration and turnout in surveys: Do official government records yield more accurate assessments? *Public Opinion Quarterly*, *80*(3), 597–621.

Berlusconi, G., Calderoni, F., Parolini, N., Verani, M., & Piccardi, C. (2016). Link prediction in criminal networks: A tool for criminal intelligence analysis. *PLOS ONE*, *11*(4), 1-21.

Bhattacharya, M., & Mani, P. (2015). Preserving privacy in social network graph with k-anonymize degree sequence generation. *2015 9th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, 1–7.

Bloch, F., Jackson, M. O., & Tebaldi, P. (2019). Centrality measures in networks. *SSRN*, 1–41.

Boulianne, S. (2019). Revolution in the making? Social media effects across the globe. *Information, Communication & Society*, *22*(1), 39–54.

Bouwman, H., Nikou, S., Molina-Castillo, F. J., & De Reuver, M. (2018). The impact of digitalization on business models. *Digital Policy, Regulation and Governance*, *20*(2), 105–124.

Brandes, U. (2008). On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, *30*(2), 136–145.

Broumi, S., Bakal, A., Talea, M., Smarandache, F., & Vladareanu, L. (2016). Applying Dijkstra algorithm for solving neutrosophic shortest path problem. *International Conference on Advanced Mechatronic Systems (ICAMechS)*, 412–416.

Burcher, M., & Whelan, C. (2018). Social network analysis as a tool for criminal intelligence: Understanding its potential from the perspectives of intelligence analysts. *Trends in Organized Crime*, *21*(3), 278–294.

Cadwalladr, C., & Graham Harrison, E. (2018). Revealed: 50 million Facebook profiles harvested for Cambridge Analytica in major data breach. *The Guardian*, 1-6.

Campan, A., & Truta, T. M. (2008). Data and structural k-anonymity in social networks. *International Workshop on Privacy, Security, and Trust in KDD*, 33–54.

Casas Roma, J., Herrera Joancomartí, J., & Torra, V. (2013). An algorithm for k-degree anonymity on large networks. *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 671–675.

Causey, D., Pinder, T., & Doersam, A. (2016). Data theft damages: Who pays? *ABA Banking Journal*, *108*(5), Article#18.

Chen, L., & Zhu, P. (2015). Preserving the privacy of social recommendation with a differentially private approach. *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, 780–785.

Cheng, J., Fu, A. W. C., & Liu, J. (2010). k-isomorphism: Privacy preserving network publication against structural attacks. *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, 459–470.

Clement, J. (Feb 14, 2020). *Most popular social networks worldwide as of January 2020, ranked by number of active users (in millions).* Retrieved on March 23, 2020, from https://www.statista.com/statistics/272014/global-social-networks-ranked-by-number-of-users/

Cormode, G., Srivastava, D., Yu, T., & Zhang, Q. (2008). Anonymizing bipartite graph data using safe groupings. *Proceedings of the VLDB Endowment*, *1*(1), 833–844.

D'Agostino, R. (2017). *Goodness-of-fit-techniques*. CRC Press.

Das, K., Samanta, S., & Pal, M. (2018). Study on centrality measures in social networks: A survey. *Social Network Analysis and Mining*, *8*(1), Article#13.

Das, S., Eğecioğlu, Ö., & El Abbadi, A. (2010a). Anónimos: An LP-based approach for anonymizing weighted social network graphs. *IEEE Transactions on Knowledge and Data Engineering*, *24*(4), 590–604.

Das, S., Eğecioğlu, Ö., & El Abbadi, A. (2010b). Anonymizing weighted social network graphs. *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, 904–907.

Day, W. Y., Li, N. h., & Lyu, M. (2016). Publishing graph degree distribution with node differential privacy. *Proceedings of the 2016 International Conference on Management of Data*, 123–138.

Deng, Y., Chen, Y., Zhang, Y., & Mahadevan, S. (2012). Fuzzy dijkstra algorithm for shortest path problem under uncertain environment. *Applied Soft Computing*, *12*(3), 1231–1237.

Di Gangi, P. M., & Wasko, M. M. (2016). Social media engagement theory: Exploring the influence of user engagement on social media usage. *Journal of Organizational and End User Computing (JOEUC)*, *28*(2), 53–73.

Doane, D. P., & Seward, L. E. (2011). Measuring skewness: A forgotten statistic? *Journal of Statistics Education*, *19*(2), 1–17.

Dokov, S., Morton, D. P., & Popova, I. (2017). Mean-variance-skewness-kurtosis efficiency of portfolios computed via moment-based bounds. *International Conference on Information Science and Communications Technologies (ICISCT)*, 1–5.

Doric, D., Nikolic Doric, E., Jevremovic, V., & Malislic, J. (2009). On measuring skewness and kurtosis. *Quality and Quantity*, *43*(3), 481–493.

Dwork, C. (2008). Differential privacy: A survey of results. *International Conference on Theory and Applications of Models of Computation*, 1–19.

Fard, A. M., & Wang, K. (2015). Neighborhood randomization for link privacy in social network analysis. *World Wide Web*, *18*(1), 9–32.

Finner, H., & Gontscharuk, V. (2018). Two-sample Kolmogorov-Smirnov-type tests revisited: Old and new tests in terms of local levels. *The Annals of Statistics*, *46*(6A), 3014–3037.

Fukuta, Y., Murata, K., Adams, A. A., Orito, Y., & Palma, A. M. L. (2017). Personal data sensitivity in Japan: An exploratory study. *ORBIT Journal*, *1*(2), 1-13.

Gan, M. F., Chua, H. N., & Wong, S. F. (2018). Personal Data Protection Act enforcement with PETs adoption: An exploratory study on employees' working process change. In *IT Convergence and Security 2017* (pp. 193–202). Springer.

Gong, M., Li, G., Wang, Z., Ma, L., & Tian, D. (2016). An efficient shortest path approach for social networks based on community structure. *CAAI Transactions on Intelligence Technology*, *1*(1), 114–123.

Hay, M., Li, C., Miklau, G., & Jensen, D. (2009). Accurate estimation of the degree distribution of private networks. *2009 Ninth IEEE International Conference on Data Mining*, 169–178.

Hay, M., Miklau, G., Jensen, D., Srivastava, S., & Weis, P. (2007). *Anonymizing social networks* (Tech. Rep.). Computer Science Department Faculty Publication Series.

Hay, M., Miklau, G., Jensen, D., Towsley, D., & Weis, P. (2008). Resisting structural re-identification in anonymized social networks. *Proceedings of the VLDB Endowment*, *1*(1), 102–114.

Huang, D. W., Yu, Z. G., & Anh, V. (2017). Multifractal analysis and topological properties of a new family of weighted koch networks. *Physica A: Statistical Mechanics and its Applications*, *469*, 695–705.

Isaak, J., & Hanna, M. J. (2018). User data privacy: Facebook, Cambridge Analytica, and privacy protection. *Computer Society*, *51*(8), 56–59.

Jan, S. K., & Vlachopoulos, P. (2019). Social network analysis: A framework for identifying communities in higher education online learning. *Technology, Knowledge and Learning*, *24*(4), 621–639.

Jones, C. I., & Tonetti, C. (2019). *Nonrivalry and the economics of data* (Tech. Rep.). National Bureau of Economic Research.

Kalia, P., Arora, R., & Law, P. (2017). Information Technology Act in India: E-commerce value chain analysis. *NTUT Journal of Intellectual Property Law and Management*, *5*(2), 55–97.

Kotani, H., & Yokomatsu, M. (2018). Quantitative evaluation of the roles of community events and artifacts for social network formation: A multilayer network model of a community of practice. *Computational and Mathematical Organization Theory*, *25*, 1–36.

Kumbhojkar, P., Jain, M., Rajalakshmi, E., Rawal, S., & Thombre, S. (2018). Interface implementation for quantifying information spread on social networks. *2018 IEEE 6th International Conference on MOOCs, Innovation and Technology in Education (MITE)*, 48–51.

Lawrenz, S., Sharma, P., & Rausch, A. (2019). Blockchain technology as an approach for data marketplaces. *Proceedings of the 2019 International Conference on Blockchain Technology*, 55–59.

Lee, H., Choi, J., Kim, K. K., & Lee, A. R. (2014). Impact of anonymity on information sharing through internal psychological processes: A case of south korean online communities. *Journal of Global Information Management (JGIM)*(3), 57–77.

LeFevre, K., DeWitt, D. J., & Ramakrishnan, R. (2005). Incognito: Efficient full-domain k-anonymity. *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, 49–60.

Li, X., Yang, J., Sun, Z., & Zhang, J. (2017). Differential privacy for edge weights in social networks. *Security and Communication Networks*, *2017*, 1–10.

Liu, K., & Terzi, E. (2008). Towards identity anonymization on graphs. *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, 93–106.

Liu, L., Liu, J., Zhang, J., & Wang, J. (2010). Privacy preservation of affinities in social networks. *Proceedings of the International Conference on Information Systems*, 372–376.

Liu, L., Wang, J., Liu, J., & Zhang, J. (2008). *Privacy preserving in social networks against sensitive edge disclosure* (Tech. Rep.). CMIDA-HiPSCCS 006-08, Department of Computer Science, University of Kentucky, KY.

Liu, L., Wang, J., Liu, J., & Zhang, J. (2009). Privacy preservation in social networks with sensitive edge weights. *Proceedings of the 2009 SIAM International Conference on Data Mining*, 954–965.

Liu, M., Zeng, Y., Jiang, Z., Liu, Z., & Ma, J. (2017). Centrality based privacy preserving for weighted social networks. *13th International Conference on Computational Intelligence and Security (CIS)*, 574–577.

Liu, P., Wang, L. E., & Li, X. (2017). Randomized perturbation for privacy-preserving social network data publishing. *2017 IEEE International Conference on Big Knowledge (ICBK)*, 208–213.

Liu, P., Xu, Y., Jiang, Q., Tang, Y., Guo, Y., Wang, L., & Li, X. (2019). Local differential privacy for social network publishing. *Neurocomputing*, *391*, 273-279.

Liu, Q., Wang, G., Li, F., Yang, S., & Wu, J. (2016). Preserving privacy with probabilistic indistinguishability in weighted social networks. *IEEE Transactions on Parallel and Distributed Systems*, *28*(5), 1417–1429.

Liu, S., Zhang, D. G., Liu, X. H., Zhang, T., Gao, J. X., Cui, Y. Y., & Chang, l. G. (2019). Dynamic analysis for the average shortest path length of mobile ad hoc networks under random failure scenarios. *IEEE Access*, *7*, 21343–21358.

Lu, F. S., Hattab, M. W., Clemente, C. L., Biggerstaff, M., & Santillana, M. (2019). Improved state-level influenza nowcasting in the United States leveraging Internet-based data and network approaches. *Nature Communications*, *10*(1), 1–10.

Lu, Q., Myers, A., & Beliveau, S. (2017). USPTO patent prosecution research data:

Unlocking office action traits. *USPTO Economic Working Paper*, *10*, 1–41.

Machanavajjhala, A., Gehrke, J., Kifer, D., & Venkitasubramaniam, M. (2006). l-diversity: Privacy beyond k-anonymity. *22nd International Conference on Data Engineering (ICDE 2006)*, 24–24.

Macwan, K. R., & Patel, S. J. (2018). Node differential privacy in social graph degree publishing. *Procedia Computer Science*, *143*, 786–793.

Martens, T. (2019). The disclosure function of the US patent system: Evidence from the US Patent and Trademark Depository Library program. *SSRN*, 1–44.

Maslov, S., & Sneppen, K. (2002). Specificity and stability in topology of protein networks. *Science*, *296*(5569), 910–913.

Massey Jr, F. J. (1951). The Kolmogorov-Smirnov test for goodness of fit. *Journal of the American Statistical Association*, *46*(253), 68–78.

McCallister, E., Grance, T., & Scarfone, K. A. (2010). *SP 800-122. Guide to protecting the confidentiality of personally identifiable information (PII)* (Tech. Rep.). National Institute of Standards & Technology.

McGlohon, M., Akoglu, L., & Faloutsos, C. (2011). Statistical properties of social networks. In *Social Network Data Analytics* (pp. 17–42). Springer.

Merivaki, T., & Smith, D. A. (2020). Challenges in Voter Registration. In *The Future of Election Administration* (pp. 59–82). Springer.

Newman, R., Chang, V., Walters, R. J., & Wills, G. B. (2016). Web 2.0-The past and the future. *International Journal of Information Management*, *36*(4), 591–598.

Nissim, K., Raskhodnikova, S., & Smith, A. (2007). Smooth sensitivity and sampling in private data analysis. *Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing*, 75–84.

Novak, A. N., & Vilceanu, M. O. (2019). "The Internet is not pleased": Twitter and the 2017 Equifax data breach. *The Communication Review*, *22*(3), 196–221.

O'Dea, S. (Feb 28, 2020). *Volume of data/ information created worldwide from 2010 to 2025 (in zetabytes).* Retrieved on March 23, 2020, from https://www.statista.com/statistics/871513/worldwide-data-created/

Opsahl, T., & Panzarasa, P. (2009). Clustering in weighted networks. *Social Networks*, *31*(2), 155–163.

Park, S. B., Ok, C. M., & Chae, B. K. (2016). Using Twitter data for cruise tourism marketing and research. *Journal of Travel & Tourism Marketing*, *33*(6), 885–898.

Peng, L., Bai, Y., Wang, L., & Li, X. (2017). Partial k-anonymity for privacy-preserving social network data publishing. *International Journal of Software Engineering and Knowledge Engineering*, *27*(01), 71–90.

Peng, S., Zhou, Y., Cao, L., Yu, S., Niu, J., & Jia, W. (2018). Influence analysis in social networks: A survey. *Journal of Network and Computer Applications*, *106*, 17–32.

Pettigrew, S., & Stewart III, C. (2017). Moved out, moved on: Assessing the effectiveness of voter registration list maintenance. *MIT Political Science Department Research Paper*, *2018*(1), 1–37.

Pfitzmann, A., & Hansen, M. (2010). *A terminology for talking about privacy by data minimization: Anonymity, unlinkability, undetectability, unobservability, pseudonymity, and identity management.* Retrieved on December 17, 2020, from http://dud.inf.tu-dresden.de/Anon_Terminology.shtml

Rastogi, V., Hay, M., Miklau, G., & Suciu, D. (2009). Relationship privacy: Output perturbation for queries with joins. *Proceedings of the Twenty-Eighth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, 107–116.

Rodrigues, F. A. (2019). Network centrality: An introduction. In *A mathematical modeling approach from nonlinear dynamics to complex systems* (pp. 177–196). Springer International Publishing.

Rodriguez-Garcia, M., Batet, M., & Sánchez, D. (2019). Utility-preserving privacy protection of nominal data sets via semantic rank swapping. *Information Fusion*, *45*, 282–295.

Saoud, B., & Moussaoui, A. (2016). Community detection in networks based on minimum spanning tree and modularity. *Physica A: Statistical Mechanics and its Applications*, *460*, 230–234.

Saura, J. R., Palos Sanchez, P. R., & Correia, M. B. (2019). Digital marketing strategies based on the e-business model: Literature review and future directions. In *Organizational Transformation and Managing Innovation in the Fourth Industrial Revolution* (pp. 86–103). IGI Global.

Shen, G. C. C., Chiou, J. S., Hsiao, C. H., Wang, C. H., & Li, H. N. (2016). Effective marketing communication via social networking site: The moderating role of the social tie. *Journal of Business Research*, *69*(6), 2265–2270.

Skarkala, M. E., Maragoudakis, M., Gritzalis, S., Mitrou, L., Toivonen, H., & Moen, P. (2012). Privacy preservation by k-anonymization of weighted social networks. *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, 423–428.

Strang, A., Haynes, O., Cahill, N. D., & Narayan, D. A. (2018). Generalized relationships between characteristic path length, efficiency, clustering coefficients, and density. *Social Network Analysis and Mining*, *8*(1), Article#14.

Supriya, S., Siuly, S., Wang, H., Cao, J., & Zhang, Y. (2016). Weighted visibility graph with complex network features in the detection of epilepsy. *IEEE Access*, *4*, 6554–6566.

Sweeney, L. (2002). k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *10*(05), 557–570.

Tai, C. H., Yu, P. S., Yang, D. N., & Chen, M. S. (2011). Privacy-preserving social network publication against friendship attacks. *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 1262–1270.

Tang, J., Korolova, A., Bai, X., Wang, X., & Wang, X. (2017). Privacy loss in Apple's implementation of differential privacy on macos 10.12. *ArXiv*, *abs/1709.02753*, 1–12.

Tang, J., Zhang, S., Zhang, W., Liu, F., Zhang, W., & Wang, Y. (2016). Statistical properties of urban mobility from location-based travel networks. *Physica A: Statistical Mechanics and its Applications*, *461*, 694–707.

Tesfay, W. B., Hofmann, P., Nakamura, T., Kiyomoto, S., & Serna, J. (2018). Privacy guide: Towards an implementation of the EU GDPR on Internet privacy policy evaluation. *Proceedings of the Fourth ACM International Workshop on Security and Privacy Analytics*, 15–21.

Thiel, S., Hermann, F., Heupel, M., & Bourimi, M. (2013). Unlinkability support in a decentralised, multiple-identity social network. *Open Identity Summit*, 32–42.

Trautman, L. J., & Ormerod, P. C. (2016). Corporate directors' and officers' cybersecurity standard of care: The Yahoo data breach. *American University Law Review*, *66*, Article#1231.

Ventrella, A. V., Piro, G., & Grieco, L. A. (2018). On modeling shortest path length distribution in scale-free network topologies. *IEEE Systems Journal*, *12*(4), 3869–3872.

Villi, M., & Picard, R. G. (2019). Transformation and innovation of media business models. *Making Media: Production, Practices, and Professions*, 121–132.

Voigt, P., & Von dem Bussche, A. (2017). The EU General Data Protection Regulation (GDPR). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*.

Wallace, M. K., & Reinman, S. (2018). Expanding the intellectual property knowledge base at university libraries: Collaborating with Patent and Trademark Resource Centers. *Science and Technology Librarianship*.

Wang, S. L., Shih, C. C., Ting, I. H., & Hong, T. P. (2013). Degree anonymization for k-shortest-path privacy. *2013 IEEE International Conference on Systems, Man, and Cybernetics*, 1093–1097.

Wang, S. L., Tsai, Y. C., Kao, H. Y., Ting, I. H., & Hong, T. P. (2013). Shortest paths anonymization on weighted graphs. *International Journal of Software Engineering and Knowledge Engineering*, *23*(01), 65–79.

Wang, S. L., Tsai, Z. Z., Hong, T. P., & Ting, I. H. (2011). Anonymizing shortest paths on social network graphs. *Asian Conference on Intelligent Information and Database Systems*, 129–136.

Wilson, R. L., & Rosen, P. A. (2003). Protecting data through perturbation techniques: The

impact on knowledge discovery in databases. *Journal of Database Management (JDM)*, *14*(2), 14–26.

Xiong, F., Liu, Y., & Cheng, J. (2017). Modeling and predicting opinion formation with trust propagation in online social networks. *Communications in Nonlinear Science and Numerical Simulation*, *44*, 513–524.

Ying, X., Pan, K., Wu, X., & Guo, L. (2009). Comparisons of randomization and k-degree anonymization schemes for privacy preserving social network publishing. *Proceedings of the 3rd Workshop on Social Network Mining and Analysis*, 1–10.

Ying, X., & Wu, X. (2008). Randomizing social networks: A spectrum preserving approach. *Proceedings of the 2008 SIAM International Conference on Data Mining*, 739–750.

Yuan, M., & Chen, L. (2011). Node protection in weighted social networks. *International Conference on Database Systems for Advanced Applications*, 123–137.

Zhan, Q., Yu, T., Yang, Y., Khalil, I., Xiao, X., & Ren, K. (2017). Generating synthetic decentralized social graphs with local differential privacy. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 425–438.

Zhang, Y., Bai, Y., Chen, L., Bian, K., & Li, X. (2016). Influence maximization in messenger-based social networks. *IEEE Global Communications Conference (GLOBECOM)*, 1–6.

Zheleva, E., & Getoor, L. (2007). Preserving the privacy of sensitive relationships in graph data. *International Workshop on Privacy, Security, and Trust in KDD*, 153–171.

Zhou, B., & Pei, J. (2008). Preserving privacy in social networks against neighborhood attacks. *IEEE 24th International Conference on Data Engineering*, 506–515.

Zhou, B., & Pei, J. (2011). The k-anonymity and l-diversity approaches for privacy preservation in social networks against neighborhood attacks. *Knowledge and Information Systems*, *28*(1), 47–77.

Zhu, C. (2019). Big data as a governance mechanism. *The Review of Financial Studies*, *32*(5), 2021–2061.

Zhuang, L., Zhou, F., Zhao, B. Y., & Rowstron, A. (2005). Cashmere: Resilient anonymous routing. *Proceedings of the 2nd Conference on Symposium on Networked Systems Design & Implementation*, 301–314.

Zou, L., Chen, L., & Özsu, M. T. (2009). k-automorphism: A general framework for privacy preserving network publication. *Proceedings of the VLDB Endowment*, *2*(1), 946–957.