UNIVERSITY OF THE AEGEAN

DOCTORAL THESIS

Privacy Preserving Data Mining

---

by

Maria Eleni Skarkala

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy at the

Department of Information and Communication Systems Engineering
School of Engineering, University of the Aegean
Karlovassi, Samos
Greece

March, 2021

Copyright ©2021 Maria Eleni Skarkala

Department of Information and Communication Systems Engineering
School of Engineering
University of the Aegean

All rights reserved.

Declaration of Authorship

I, Maria Eleni Skarkala, declare that I am the sole author of this dissertation entitled, "Privacy preserving data mining"and that I have not used any sources other than those listed in the bibliography and identified as references. I further declare that I have not submitted this dissertation at any other institution in order to obtain a degree, diploma or other qualification. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Tuesday 9th March, 2021	Maria Eleni Skarkala
Date	*Signature*

Advising Committee

Manolis Maragoudakis, Supervisor
Professor, Ionian University
Greece

Stefanos Gritzalis, Advisor
Professor, University of Piraeus
Greece

Evangelia (Lilian) Mitrou, Advisor
Professor, University of the Aegean
Greece

Examining Committee

Manolis Maragoudakis
Professor, Ionian University
Greece

Stefanos Gritzalis
Professor, University of Piraeus
Greece

Evangelia (Lilian) Mitrou
Professor, University of the Aegean
Greece

Maria Karyda
Associate Professor, University of the Aegean
Greece

Christos Kalloniatis
Associate Professor, University of the Aegean
Greece

Aggeliki Tsohou
Assistant Professor, Ionian University
Greece

Katia Lida Kermanidis
Associate Professor, Ionian University
Greece

To my family.
Kiki
Dora
Panos
Spyros
Nitsa
Nikos
Eleni
Nikoleta
Giorgos
and Nektarios.

Abstract

Medical, financial, or social databases are analyzed daily for the discovery of patterns and useful information. Privacy concerns have emerged as some database segments contain sensitive data. Data mining techniques are used to parse, process, and manage enormous amounts of data while ensuring the preservation of private information, as data can be exploited by potential aggressors. Regarding social networks, their privacy preserving analysis aims to understand better the network and its behavior, while at the same time protecting the privacy and identity of its individuals. Network data contain sensitive information and due to the increasing popularity of social networks that are released publicly, effective anonymization techniques are required to make the data available for research.

Considering the above, this thesis is divided in two parts and focuses on privacy preservation of distributed databases and social network data. In the first part, a privacy preserving data mining protocol is presented, thoroughly designed and developed for both horizontally and vertically partitioned databases, which contain either nominal or numeric attribute values. At the same time the accuracy of final outcomes and the preservation of privacy is the main goal of the proposed protocol. Cryptography, as shown by previous research, is the most accurate approach to acquiring knowledge while maintaining privacy to assure both confidentiality and integrity of data. The proposed algorithm exploits the multi-candidate election schema to construct a privacy-preserving tree-augmented naive Bayesian classifier, a more robust variation of the classical naive Bayes classifier. The exploitation of the Paillier cryptosystem and the distinctive homomorphic primitive shows in the security analysis that privacy is ensured and the proposed algorithm provides strong defences against common attacks.

In the second part, an anonymization algorithm is developed for weighted graphs, i.e., for social networks where the strengths of links are important. Previous studies concentrate mainly on preventing identity disclosure in unweighted graphs. However, a weighted graph is more descriptive, revealing more information about the relationships between entities, which allows adversaries to take advantage of potential security holes. Weights can be essential for social network analysis, but they pose new challenges to privacy preserving network analysis. For instance, an adversary may use his information about some edge weights to re-identify individuals. This in contrast with many previous studies which only consider unweighted graphs. The proposed anonymization method considers identity, edge and edge weight disclosure for anonymizing weighted graph data, assuming that adversaries have knowledge about the neighborhood of a targeting entity. In particular, a k-anonymous technique is presented that groups entities with same neighborhoods into supernodes and the corresponding connections into superedges. The method provides k-anonymity of nodes against attacks where the adversary has information about the structure

of the network, including its edge weights.

Both approaches are proven efficient and have been evaluated in terms of privacy and utility. Experiments deriving the benefits of real world databases demonstrate the preservation of private data while mining processes occur.

Keywords Privacy; Data mining; Privacy preserving; Distributed databases; Social networks; Security; Privacy preserving analysis; Tree Augmented Naive Bayes; Paillier cryptosystem; Homomorphic encryption; Anonymization; k-anonymity; Generalization; Privacy Disclosure; Weighted social network; Identity disclosure; Edge disclosure; Edge weight disclosure

Περίληψη

Ιατρικές, οικονομικές ή κοινωνικές βάσεις δεδομένων αναλύονται καθημερινά για την ανακάλυψη προτύπων και χρήσιμων πληροφοριών. Προβλήματα απορρήτου προκύπτουν καθώς ορισμένα τμήματα βάσης δεδομένων περιέχουν ευαίσθητα δεδομένα. Οι τεχνικές εξόρυξης δεδομένων χρησιμοποιούνται για την ανάλυση, την επεξεργασία και τη διαχείριση τεράστιων ποσοτήτων δεδομένων, διασφαλίζοντας παράλληλα τη διατήρηση των ιδιωτικών πληροφοριών, καθώς τα δεδομένα μπορούν να αξιοποιηθούν από πιθανούς επιτιθέμενους. Όσον αφορά τα κοινωνικά δίκτυα, η ανάλυσή τους για την προστασία της ιδιωτικότητας στοχεύει στην καλύτερη κατανόηση του δικτύου και της συμπεριφοράς του, ενώ ταυτόχρονα προστατεύει το απόρρητο και την ταυτότητα των ατόμων του. Τα δεδομένα δικτύου περιέχουν ευαίσθητες πληροφορίες και λόγω της αυξανόμενης δημοτικότητας τους που δημοσιεύονται δημοσίως, απαιτούνται αποτελεσματικές τεχνικές ανωνυμοποίησης για τη διάθεση των δεδομένων για έρευνα.

Λαμβάνοντας υπόψη τα παραπάνω, αυτή η διατριβή χωρίζεται σε δύο μέρη και εστιάζει στη διατήρηση της ιδιωτικότητας σε κατανεμημένες βάσεις δεδομένων και δεδομένα κοινωνικών δικτύων. Στο πρώτο μέρος, παρουσιάζεται ένας αλγόριθμος εξόρυξης δεδομένων διατήρησης απορρήτου, σχεδιασμένος και υλοποιημένος διεξοδικά για οριζόντιες και κατακόρυφες κατατμημένες βάσεις δεδομένων, οι οποίες περιέχουν χαρακτηριστικά είτε με ονομαστικές είτε με αριθμητικές τιμές. Ταυτόχρονα, η ακρίβεια των τελικών αποτελεσμάτων και η διατήρηση της ιδιωτικότητας είναι ο κύριος στόχος του προτεινόμενου πρωτοκόλλου. Η κρυπτογραφία, όπως φαίνεται από προηγούμενη έρευνα, είναι η πιο ακριβής προσέγγιση για την απόκτηση γνώσεων, διατηρώντας παράλληλα το απόρρητο για να διασφαλιστεί τόσο η εμπιστευτικότητα όσο και η ακεραιότητα των δεδομένων. Ο προτεινόμενος αλγόριθμος εκμεταλλεύεται το πολυυποψήφιο σχήμα εκλογής για να κατασκευάσει ένα tree-augmented naive Bayesian ταξινομητή, μια πιο ισχυρή παραλλαγή του κλασικού αφελής ταξινομητή Bayes. Η εκμετάλλευση του κρυπτοσυστήματος Paillier και η θεμελιώδης ομομορφική αρχή δείχνουν στην ανάλυση ασφάλειας ότι διασφαλίζεται η προστασία της ιδιωτικότητας και ο προτεινόμενος αλγόριθμος παρέχει ισχυρές άμυνες ενάντια σε κοινές επιθέσεις.

Στο δεύτερο μέρος, αναπτύσσεται μια μέθοδος ανωνυμοποίησης για σταθμισμένα γραφήματα, δηλαδή για κοινωνικά δίκτυα όπου η ισχύς των συνδέσμων είναι σημαντική. Προηγούμενες μελέτες επικεντρώνονται κυρίως στην αποτροπή της αποκάλυψης ταυτότητας σε μη σταθμισμένα γραφήματα. Ωστόσο, ένα σταθμισμένο γράφημα είναι πιο περιγραφικό, αποκαλύπτοντας περισσότερες πληροφορίες σχετικά με τις σχέσεις μεταξύ οντοτήτων, γεγονός που επιτρέπει στους επιτιθέμενους να επωφεληθούν από πιθανές τρύπες ασφαλείας. Τα βάρη μπορούν να είναι απαραίτητα για την ανάλυση κοινωνικών δικτύων, αλλά θέτουν νέες προκλήσεις στην προστασία του απορρήτου για την ανάλυση δικτύων. Για παράδειγμα, ένας επιτιθέμενος μπορεί να χρησιμοποιήσει τις πληροφορίες του σχετικά με κάποια βάρη συνδέσμων για να επαναπροσδιορίσει τα άτομα. Αυτό έρχεται σε αντίθεση με πολλές προηγούμενες μελέτες που θεωρούν μόνο

μη σταθμισμένα γραφήματα. Η προτεινόμενη μέθοδος ανωνυμοποίησης λαμβάνει υπόψη την ταυτότητα, την σύνδεση και το βάρος της σύνδεσης για την ανωνυμοποίηση σταθμισμένων δεδομένων γραφήματος, υποθέτοντας ότι οι επιτιθέμενοι έχουν γνώση σχετικά με τη γειτονιά μιας στοχευμένης οντότητας. Συγκεκριμένα, παρουσιάζεται μια k - ανώνυμη τεχνική που ομαδοποιεί οντότητες με τις ίδιες γειτονιές σε υπεροντότητες και τις αντίστοιχες συνδέσεις σε υπερσυνδέσεις. Η μέθοδος παρέχει k - ανωνυμία κόμβων έναντι επιθέσεων όπου ο επιτιθέμενος έχει πληροφορίες σχετικά με τη δομή του δικτύου, συμπεριλαμβανομένων των βαρών.

Και οι δύο μεθοδολογίες έχουν αποδειχθεί αποτελεσματικές και έχουν αξιολογηθεί ως προς το απόρρητο και τη χρησιμότητα. Τα πειράματα που αντλούν τα οφέλη από πραγματικές βάσεις δεδομένων δείχνουν τη διατήρηση των ιδιωτικών δεδομένων κατά τη διάρκεια τεχνικών εξόρυξης γνώσης.

Λέξεις-κλειδιά Απόρρητο; Εξόρυξη δεδομένων; Διατήρηση της ιδιωτικότητας; Κατανεμημένες βάσεις δεδομένων; Κοινωνικά δίκτυα; Ασφάλεια; Ανάλυση προστασίας της ιδιωτικότητας; Tree Augmented Naive Bayes; Κρυπτοσύστημα Paillier; Ομομορφική κρυπτογράφηση; Ανωνυμοποίηση; k -ανωνυμία; Γενίκευση; Αποκάλυψη απορρήτου; Σταθμισμένο κοινωνικό δίκτυο; Αποκάλυψη ταυτότητας; Αποκάλυψη συνδέσμου; Αποκάλυψη βάρους σύνδεσης

Acknowledgements

The journey of my doctoral research has not been easy, but many people throughout all these years were there to encourage me.

Firstly, I would like to express my gratitude to my supervisors and mentors, Prof. Manolis Maragoudakis, Prof. Stefanos Gritzalis and Prof. Lilian Mitrou for their continuous support during this journey. Their patience and guidance encouraged me on continuing and finally completing my dissertation.

I would also like to thank my professors at the Helsinki University, Prof. Hannu Toivonen and Pirjo Moen, for their support and advice while being a part of the Algorithmic Data Analysis (Algodan) Center of Excellence of the Academy of Finland.

Furthermore, I would like to thank all the members of the examination committee for taking the time to review the thesis and provide useful advice.

A special thank you to my family, and especially my mother Kiki, who is always by my side and supports all my decisions. She is the reason why I kept working hard for my goals and her encouragement gave me the strength throughout my research. A second special thank you goes to my grandmother, Eleni, who is always asking me how my progress is going, and always pushing me to continue to overcome my goals.

To my friends, a huge thank you for pushing me to finish my research, so that they will have a friend to call "Dr.". Thank you for your faith in me.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Overview

The advancement of the Internet and technologies have increased the amounts of data in various fields, and data applications have evolved from simple storage to acquiring knowledge. Therefore the process of revealing important hidden patterns and associations from a dataset, called data mining, advanced over the years. Data mining has a wide range of applications and plays an essential role as through operations and techniques automatic and algorithmic tools are created to generate useful information and knowledge from data. For example, governments apply data mining techniques to gain insights on citizens characteristics and companies to know how their customers behave.

Data mining techniques are the main tools to extract knowledge. Generally, data mining methods are categorized in three types: classification, clustering and association rule mining. Classification methods are supervised learning techniques in which classes are pre-determined. Clustering methods are unsupervised learning methods and are not predefined. Association rule mining methods searches for interesting relations in a dataset.

Acquiring knowledge though in many cases can violate the privacy of the individuals involved, and oppositely privacy poses restrictions to accessing knowledge. Balancing access to knowledge and preserving privacy at the same time poses many challenges. Thus, both data mining and information security research communities are interested in overcoming this obstacle.

Social networks are a modern concept that have gained popularity in recent years. That increased the network data that have been publicly available. Therefore, the utility of social networks has extended beyond the user's activity, as researchers analyze network data to extract valuable information. Due to their nature, privacy concerns have also been raised in this area. Data owners must protect the users who are related with these data before releasing the datasets to the public. Thus, it is important to provide methods that can efficiently hide sensitive information and ensure anonymity.

In machine learning and privacy-preserving data mining there are two main concepts that should be considered. The first focuses on anonymizing data and suppressing identifiers to preserve privacy. The second refers to protecting privacy of collections of datasets by securing data from unauthorized access. This dissertation provides solutions for both dimensions.

Data, in some cases, are distributed in multiple parties, and different organizations need to collaborate to run a data mining algorithm on their dataset union. Standard data mining algorithms do not run on distributed data therefore their modification is required. In real world applications, privacy issues arise due to security and legal constraints as parties cannot simply send their data to a third party to execute data mining methods and acquire knowledge. Therefore, the increasing demands on protecting privacy and sensitive information created the need to develop privacy preserving data mining (PPDM) techniques.

Since 2000, the research in this field has increased dramatically, and many algorithms and protocols have been proposed for different data mining techniques. Privacy preserving data mining aims to solve the problem of protecting privacy of the individuals involved in data mining operations [1].

In distributed databases, the problem is defined as parties jointly conduct a data mining process with their private data sets as input. For example, several medical institutions wish to collaborate with each other, giving their data as input, such that a data mining operation is performed, while privacy is preserved, to extract knowledge i.e. if a patient will develop a disease based on their medical history. When the operation is complete each party knows only their local data and the global results of the mining process. Based on Goldreich [2], privacy is achieved if each party can have access only to the local input and output of the process. This definition is also used for anonymity of sensitive data and the owner identity. However, Backstorm et al [3] show that the technique of simply removing identifiers does not guarantee privacy. Therefore, more advanced anonymization methods are required to achieve protection of individuals' identity and sensitive information.

There are two main approaches for privacy preservation over distributed databases: secure multi-party computation (SMC) and randomization. In secure multi-party computation, the final outcomes are computed using cryptographic tools among two or more parties who jointly compute a function with private data as input. In randomization, data are perturbed using randomization and perturbation techniques before sending them and their reconstruction occurs at the final destination of the data.

A social network and every network can be modeled as a graph which may include additional information about the involved individuals and their relationships, such as the strength of their connections which is represented by weights. For example, a network of Twitter users is represented by a graph, where nodes are considered the users, edges are the in-between connections and the weights show how often the users communicate with each other. Privacy preservation in social networks can be achieved by using either clustering-based or modification methods. In clustering-based approaches, social data are clustered or generalized into groups. In modification approaches, the graph is modified by inserting or deleting data (nodes and/or edges).

Information loss measures the quantity of damage on the original data. For example, by adding noise on the original data to preserve privacy can result in data which are no longer useful to extract knowledge. Or, in social networks, the anonymization method can mask the data such that graph properties are not able to be obtained after the process. Balancing the data privacy and data utility is challenging, considering at the same time the background knowledge of an adversary.

This dissertation focuses in two main topics and methods were designed and

developed for: (1) privacy preservation of distributed databases and (2) privacy preservation of social networks. Data utility and information loss is one of the two main objectives. The second and most important objective is the privacy preservation of sensitive data and individual's identity. The methods described in this dissertation are developed such that there is a balance between privacy and utility.

1.2 Problem description

1.2.1 Privacy and security

Everyday the world is getting more digitized, increasing the electronic data which can be analyzed to discover knowledge, such as social and economic trends [4]. As these data can be disclosed to unauthorized people, privacy becomes an important aspect. Privacy can be defined as the prevention of sensitive information disclosure when data mining is performed.

Both privacy and security put constraints in data mining tasks. Privacy and security are two terms used interchangeably under different contexts, but both are related to each other. Security is the process which needs to be implemented to ensure privacy. Confidentiality, integrity and availability are the three fundamentals of security. Security can be accomplished through controls of accessing an individual's information and protect it from unauthorized disclosure, modification, loss or destruction. Security establishes policies and processes to obtain privacy and confidentiality, including integrity mechanisms that safeguard information from unauthorized modification. Privacy is defined as the right of an individual to keep his personal information secret and not disclosed. Individual's personal information may lead to his identification if disclosed. Privacy can be accomplished through policies and procedures.

For example, medical data are sensitive as they contain information about the patients identities and their diseases. Analyzing medical data can help in generating knowledge on how diseases are created or evolved. Therefore, the data need to be anonymized before publicly released for data mining purposes. It is important though to preserve privacy utilizing policies with efficient mining methods, so that the utility is not reduced, and does not lead to inaccurate results which in case of medical data can result in wrong predictions that are unacceptable.

1.2.2 Privacy preserving data mining framework

A general privacy preserving data mining framework is presented in Figure 1.1. In data mining processes the data is collected by a single or multiple parties / organizations and stored at databases. Privacy should be considered even at this stage [5].

The data is transformed and sanitized for analysis purposes. Data anonymization is performed by the data owner or a trusted party, in order to prevent sensitive information disclosure. The processes applied are blocking, suppression, perturbation, modification, generalization, sampling etc. Then, privacy preserving data mining algorithms are applied for the generation of knowledge. The privacy preservation techniques should also quantify data privacy and utility.

The discovery of knowledge leads to the final results, rules and patterns that can be used for further analysis. These results can also threaten privacy. For example patterns may be revealed that can uniquely identify individuals. However, anonymization methods that protect privacy of mining processes are not yet developed [6].

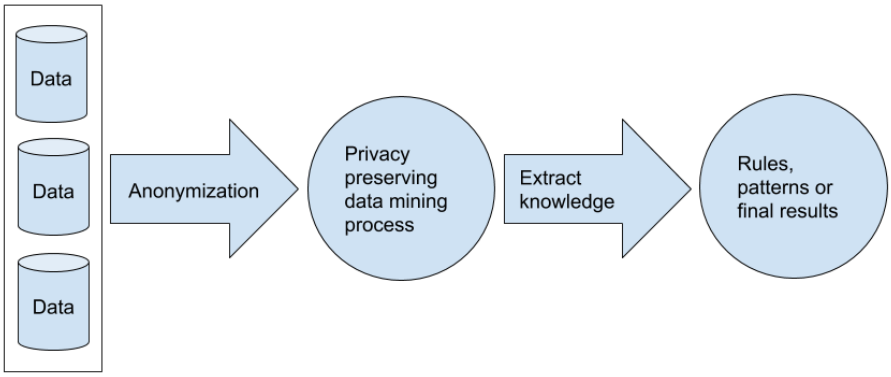


Figure 1.1: Privacy preserving data mining framework.

1.2.3 Policies and regulations

Nowadays, people are more concerned about their privacy and the privacy of their data. Due to these concerns many counties establish new privacy regulations and laws, such as the EU General Data Protection Regulation (GDPR) [1] [7] which took effect in May 2018 and the California Consumer Privacy Act of 2018 [2], or updated their existing laws such as the Australian Privacy Regulation 2013 [3] under the Privacy Act 1988 [6, 8, 9].

Guidances for health related information in the US is provided by the Health Insurance Portability and Accountability Act (HIPAA) Privacy Rule [4] [6]. Similar guidelines are provided by the Office of Australian Information Commissioner in Australia. Moreover, in Australia, people are able to opt-out of their electronic health record and delete them permanently as stated in My Health Records Amendment (Strengthening Privacy) Bill 2018 [5]. Information Technology Rules (2011) under the Information Technology Act (2000) were also introduced in India [6] to define security practices and procedures [4]. Public health data are typically protected by replacing sensitive attribute values. For example, Dutch hospitals use treatment codes. The MIMIC dataset [7], an openly available dataset developed by the MIT Lab for Computational Physiology uses subject IDs and removes sensitive attributes from data [6].

Industry conventions, besides regulations, are also mandatory. Different companies should agree on how sensitive data is collected, stored and analyzed for building

[1] https://gdpr-info.eu/

[2] https://oag.ca.gov/privacy/ccpa

[3] https://www.oaic.gov.au/privacy/the-privacy-act/privacy-regulations

[4] https://www.cdc.gov/phlp/publications/topic/hipaa.html

[5] https://www.myhealthrecord.gov.au/about/legislation-and-governance/summary-privacy-protections

[6] https://www.wipo.int/edocs/lexdocs/laws/en/in/in098en.pdf

[7] https://mimic.physionet.org/

data mining applications, preserving privacy at the same time. Last but not least, education on public awareness of information security needs to be increased [10].

1.2.4 Case studies and applications

Privacy preservation has been applied in many case studies and applications. Several studies [8,11,12] have pointed out some of these cases. Kenthapadi et al [8] focus on applications in the industry by presenting case studies from companies like Google, Apple, LinkedIn, and Microsoft.

Google's RAPPOR (Randomized Aggregatable Privacy-Preserving Ordinal Response) project [13] is the first large scale deployment of differential privacy in industry. RAPPOR permits statistics to be collected for efficient, high utility analysis of the collected data with strong privacy guarantees. RAPPOR is an open-source project which can be used to improve browser security, find bugs, and provide better overall user experience.

Apple [14] uses techniques from sketching and data streaming literature and adapts them to ensure differential privacy deployed for iOS and macOS. The technology has been used for applications such as learning new words from user keyboards, learning health analytics, and device telemetry.

LinkedIn Salary is a web-scale crowdsourcing system for secure collection and presentation of compensation insights to job seekers [15]. The key idea in this system is to use noise generation and perform post-processing to achieve data consistency.

Microsoft's local differential privacy algorithm has been deployed across millions of Windows devices to collect application usage statistics in a privacy-preserving manner [16].

Privacy preserving data mining applications can be grouped in four categories: cloud computing, e-health, wireless sensor networks, and location-based services [12]. On the other hand, in [11] the authors mention different applications of privacy preserving data mining methods, such as homeland security, bioterrorism, medical database mining etc.

Cloud computing Cloud is a distributed infrastructure that collects, stores and analyses large data with great storage and computation capabilities that can be accessible through the network [12]. Individuals need to trust the cloud providers with their data and for that reason privacy-preserving techniques focus on preserving privacy in cloud-based services. Different methods have been proposed either for horizontally or vertically partitioned data stored encrypted in the cloud where queries are allowed, or publishing data to the cloud based on the k-anonymity concept.

Medical database mining One of the most sensitive data contained within databases are the health records, therefore privacy-preserving approaches must be considered in order to protect the privacy of the patients. These approaches are divided in cryptographic and non-cryptographic [12].

Databases of DNA data are growing fast due to the advances in the DNA science and forensic analysis. These data are extremely sensitive as they contain almost unique identification information about an individual. Anonymity can be protected by applying k-anonymity mechanisms, so that each record is not distinguished from at least $k-1$ other records [11].

Another application of medical database mining is bioterrorism, where medical

data are analyzed and privacy needs to be preserved for data mining purposes [11].

Wireless sensor networks Wireless sensor networks are networks of distributed sensors that monitor the physical environment [12]. Data is wirelessly exchanged between these sensors, and due to low battery capacity it is important to process data and communicate efficiently. These data may be considered sensitive, as they might be used by attackers if they are used in a house, and track the behaviour within it.

Location-based services Through a global positioning system (GPS) location information can be obtained to provide users with useful services. This information though is sensitive as an attacker might obtain it and discover the specific locations of the target entity, such as the home address, the work address etc. Also these data contain unique characteristics as they create behavioural patterns. For that reason these data are considered sensitive. Thus, privacy preservation is required in order to protect users from being tracked. Anonymity, perturbation and specific location queries can be applied to preserve privacy in location-based data [12].

Homeland security The necessity of privacy-preserving techniques for homeland security has created a number of applications because of the nature of surveillance, in order to prevent user privacy violation. Some examples of such applications are the credential validation problem, web camera surveillance, video surveillance and the watch list problem [11].

1.2.5 Challenges

Data mining can be very valuable to many applications for information discovery, but there is a growing concern about the usage of the data discovered and the privacy threats that arise by data mining operations [17]. Individuals are worried about their privacy being violated by unauthorized access to their personal data, and the purposes for which data has been collected [10]. The pioneer work of Agrawal et al. [18] and Lindell and Pinkas [19] open the field to many studies related to privacy preserving data mining in order to address the privacy concerns while knowledge is discovered by data mining operations. Many methods such as anonymization, randomization, cryptography etc are applied on data in order to preserve privacy [20]. However, these methods may generate information loss to some extent, or create computational overhead [4].

Research and industry focus on improving security of infrastructures to protect data. This however can be challenging as the concerns on data security are evolving [21]. The three most important requirements that need to be fulfilled are the data confidentiality, integrity and availability. However new requirements emerge such as data quality, completeness, timeliness etc. If the quality of data is increased, data is much more valuable, which may increase the possibility that an adversary might gain unauthorized access and violate privacy or corrupt data. Since the amount of data available is growing fast, more scalable mechanisms are required.

In many businesses, employees can have access to sensitive information. Their actions of accessing private data though can be detected within a company, and face disciplinary actions. However, most companies use internet based data management services, in which users that are not their employees, might have access to the sensitive data, resulting in more challenges as companies are not aware of the users who accessed the data. For that reason organizations need to guarantee that they

comply with security and privacy policies and requirements. Data privacy in cloud computing is a major challenge, as the user does not have control of their data. It is hard to determine where the data is stored and where it is processed.

Mining data produced by people related to their everyday life can violate privacy. For example the data from mobile devices, such as their location, can be combined with external information and uniquely identify a person from a large set.

Social networks as well have raised enormous challenges, as privacy and social network goals are two completely opposite concepts. Many techniques have been applied for anonymization of such networks which are described in Part II. User education is crucial as well to avoid many privacy breaches, by taking advantage of the privacy settings provided by the social networks [22].

Bertino et al. [23] propose that the goals of a privacy preserving data mining algorithm should be defined before evaluating the algorithm. Based on their propositions, a privacy preserving data mining algorithm should satisfy few important requirements. A privacy preserving data mining algorithm:

- "should prevent the discovery of sensitive information."
- "should be resistant to various data mining techniques."
- "should not compromise the access and the use of non sensitive data."
- "should be usable on large amounts of data."
- "should not have exponential computational complexity."
- "should not consume high amounts of resources."

However, not all privacy preserving data mining algorithms proposed in the literature satisfy all these requirements. For example, not all the proposed algorithms are resistant to various data mining methods.

1.3 Motivation and objectives

Privacy of sensitive information in real-world applications is required because of regulations and laws, but also because of the interests of businesses and institutions. For example, hospitals conduct research on medical data to acquire knowledge for faster diagnosis in emergency cases. For marketing purposes, businesses and companies are interested in knowing the habits of customers. In both examples, the private data must be protected and the individual's personal information should remain hidden by unauthorized parties. The sensitive data should be anonymized and/or distributed in a secure way such that useful information is extracted but privacy is preserved.

The privacy preserving data mining field has made progress over the years. This evolving field has gained interest from various communities such as information security and data mining. Many achievements in the field have been accomplished, however there are still many open issues.

Different data mining applications are designed for addressing different problems. There are no specific approaches designed that can be generalized and used in

different scenarios. Developing data mining protocols that are efficient, effective, secure and accurate is considered challenging, balancing at the same time the trade-off between privacy, efficiency, utility and complexity.

After reviewing the existing privacy preservation methods proposed in this field, most methodologies focus on centralized scenarios or the distributed ones are not designed to address standardization issues. They do not support different data types, and mainly concentrate on one distribution database type, horizontal or vertical. Proposed methodologies lack implementation and the approaches proposed in the literature are designed for addressing specific problems.

Some of these methodologies have been used for the anonymization of tabular data. However they are not appropriate for graph data. Even more when graphs contain additional information such as the strength of the relationships among individuals. Most methods previously proposed focus mainly on unweighted graph data. These techniques cannot provide privacy in network data which contain sensitive information about their links. Background knowledge of adversaries can compromise privacy, and re-identification may be achieved.

The objective of the dissertation is two-fold:

- Develop a privacy preserving framework for horizontally and vertically partitioned databases while useful knowledge is obtained through mining processes.

- Develop an algorithm which provides anonymization in weighted graph data, considering as background information of adversaries the neighborhood structure of the graph while keeping the data utility close to the original data.

1.4 Contribution

This dissertation makes two significant contributions which are independent, but related to the privacy preserving data mining community. To overcome existing problems and cover some holes in the privacy preserving data mining research, we propose two protocols implemented for real world applications.

A well-designed framework is developed and implemented to support both horizontally and vertically distributed databases. Until now all previous researches focus on one database partition, and they lack implementation. The proposed protocol is designed and implemented for a distributed environment where a third party who acts as the data collector utilizes cryptographic mechanisms in order to perform the mining process. The main goal of the protocol is to preserve privacy while allowing the extraction of useful information. The protocol is developed such that it can be used by real world applications. The protocol is scalable and can be easily extended to support multiple data mining operations. The threat and security analysis along with the experimental results show the applicability and performance with different encryption mechanisms and different sizes of datasets.

The second contribution is related to privacy preservation in social networks. An algorithm is presented, which was implemented for real world applications and provides anonymization of weighted graph data. Previous studies focus on unweighted graphs which do not consider the connections between the individuals that represent. A more realistic approach is to consider a graph with more information about the individuals such as the relationships between them and their strength, which

is represented by weights. An information loss metric is presented which is applied to prove that the proposed anonymization algorithm is accurate and privacy is preserved. Past researches focus mainly in identity disclosure. However the proposed approach provides privacy for different types of disclosures, such as identity, edge and edge weight disclosure. The experimental results and the security analysis of the anonymization approach show that the proposed method strikes a balance between privacy and utility for real world weighted graphs.

Part I and part II present in depth the proposed methodologies and the contribution in the research community of privacy preserving data mining.

1.5 Dissertation structure

The dissertation is organized in ten chapters, separated in three main parts. The current chapter provides an introduction to privacy preserving data mining and states the problem that needs to be addressed. The concerns and challenges on the domain are presented along with the objective and motivations, including the contribution of the dissertation.

Part I: Chapter 2 presents an introduction to privacy preserving data mining in distributed databases and its dimensions. A categorization of the existing methodologies is given along with the advantages and disadvantages of each method. A synopsis of the proposal is defined in addition to the contribution of the proposed privacy preservation framework. Chapter 3 reviews the literature of privacy preserving data mining for distributed databases. Chapter 4 outlines the proposed protocol by presenting the theoretical background in Section 4.1, and describes in detail the privacy preservation protocol in Section 4.2. The proposed method was evaluated in terms of efficiency and privacy and the results are presented in Section 4.3. Furthermore, the threat model of the proposal is provided in Section 4.4, and the part is concluded with Chapter 5 which provides a summary on the topic presented in Part I.

Part II: Chapter 6 provides an introduction to privacy preserving data mining in graphs and social networks. The k- anonymity model is explained, and how the utility can be preserved. A summary of the proposed algorithm, and its contribution concludes the chapter. Chapter 7 presents the related work on unweighted and weighted graphs. Chapter 8 describes the proposed anonymization algorithm. In particular, Section 8.1 introduces the preliminaries and defines the problem which is solved by the k-anonymization algorithm described in details in Section 8.2. The evaluation of the proposed algorithm and the threat model are given in Section 8.3 and Section 8.4, respectively. Chapter 9 presents the summary of Part II.

Part III: This is the last part of the dissertation. Chapter 10 finishes the dissertation by outlining the conclusions and findings of the research and discusses the open issues. Future directions conclude the current research on privacy preserving data mining.

Appendixes: Finally, Appendix A presents in detail the algorithm given in Part II. Appendix B includes detailed tables on the related work presented in both Part I and Part II. Appendix C presents detailed images of the proposed privacy preservation technique for distributed databases. Appendix D provides a list of the publications on which this dissertation is based on that have been published in scientific journals and conference proceedings.

Part I

Privacy preservation of distributed databases

Chapter 2

Introduction

In recent years, advances in information and communication technologies have raised deep concerns about how data, and specifically private data, are processed. The development of data mining techniques has attracted considerable attention as the principal goal is to extract knowledge from data and, in the process, discover useful patterns. Useful information can be obtained from data following these steps: (1) data preprocessing, (2) data transformation, (3) data mining, and (4) pattern presentation and evaluation [10]. The information discovered can have incredible value, though serious threats to the security of the individual's private information must be eliminated. Personal data may be accessed by unauthorized parties and used for different purposes other than the original one for which data were initially collected. The privacy-preserving data mining field has emerged, focusing on solving the privacy issues facing data mining processes. Simultaneously ensuring data accuracy and protecting privacy is the main objective of PPDM.

Public awareness has forced many governments to enforce new privacy protection laws. Regulations are essential to ensure the protection of sensitive information and individual identities. Many countries have established laws on privacy protection. For example, the European Commission released the General Data Protection Regulation (GDPR) [7], which recognizes the need to facilitate the free flow of data, and unifies and promotes the protection of personal data within the European Union. The GDPR requires the implementation of appropriately designed technical measures, and systems should consider data protection to meet the Regulation's requirements.

Database owners require their data to not be misused by data mining processes and protect their privacy while their data are further analyzed [24,25]. PPDM methods have numerous applications in medical and financial fields. Some companies, for example, aim to extract knowledge on market trends in collaboration with other companies without disclosing their sensitive data due to competition reasons. Consider, for example, several distributed medical institutes desiring to perform medical research while ensuring the privacy of their patients. They wish to run a data mining algorithm on their database union to extract accurate outcomes without revealing private information. The involved parties acknowledge the importance of combining their data, mutually benefiting from their data union but none want to reveal the private data of their patients. Applying PPDM methods, important knowledge is discovered but sensitive information is unable to be extracted by unauthorized parties [26]. Sensitive data are not only limited to financial or medical data, but may

also apply to phone calls, buying patterns, and more. Individuals are not interested in sharing personal data without their consent or its sale for various purposes [9].

Databases distributed across several parties may be partitioned either horizontally [27–30] or vertically [31, 32]. In the horizontally partitioned case, each party's database contains different records with the same set of attributes. The main objective is to mine global information from the data. In the vertically partitioned case, each party's database contains different sets of attributes for the same record set [11, 33] concerning the same identity. The union of vertically partitioned datasets allows the discovery of knowledge that cannot be obtained from each individual database. A horizontally partitioned dataset example is the medical records of a patient, where the attributes associated with the patient are common for all clinics, such as the number of the insurance card, the disease, and so forth. A vertically partitioned dataset example is buying the records of a client, where each store has specific and unique user habits and different patterns are created by each store's database [12].

Cryptography, randomization, perturbation, and k-anonymity are a few of the various privacy-preserving techniques proposed in the literature. All these methods aim to prevent the possible disclosure of sensitive information to possible adversaries when data mining processes are applied for the extraction of useful information. Numerous data encryption approaches proposed in the PPDM field are based on the idea proposed by Yao [34] and extended by Goldreich [2]. Secure multiparty computation (SMC), a subfield of cryptography [2], aims to mine global information in the form of aggregate statistics. A set of parties wishes to jointly compute a function over the combination of all partitioned private data (input) of each participant. The main aim of this process is to protect local data without revealing the input to other parties. The data collector (miner), a trusted third party, performs all necessary calculations with the input of all the acquired private data of all participants. The miner, who acts as the data collector in the proposed protocol, forwards the final results to each party, with their main concern being the preservation of privacy. This process is secure if, at the end of it, neither of the parties nor the miner can obtain information other than the final outputs [35]. The basic idea is described as follows:

> "*the computation of a function that accepts as input some data is secure if at the end of the calculation process neither party knows anything but their own personal data, which constitute one of the inputs, and the final results*". [2, 34].

The proposed framework is designed and developed based on this idea, such that its usefulness can be exploited by the industry where privacy is valuable but at the same time the discovery of knowledge can assist in performing better decisions and operations. For example, medical institutes wish to cooperate in order to extract knowledge on defining, based on their symptoms, if a patient might have a rare disease. Or, if specific symptoms examined can categorize a patient to a vulnerable group, in order to be prioritized for receiving a vaccine. Another example is the insurance companies who wish to collaborate in order to extract knowledge that will assist them in deciding if a client can receive life insurance. By utilizing the proposed framework, banks can finalize their decisions if people should be considered in order to receive a loan or other benefits.

2.1 Privacy preserving data mining dimensions

Verykios et al. [32] and Sharma et al. [9], propose the following dimensions in which privacy preserving data mining techniques can be classified: (1) data mining scenario, (2) data mining tasks, (3) data distribution, (4) data types, (5) privacy definition, and (6) protection method.

Two main data mining scenarios are used for privacy preservation. In the first scenario, the datasets are released allowing unrestricted access and data modification is used to achieve privacy. In the second scenario, the datasets are not released but data mining operations are allowed, achieving privacy using cryptographic techniques.

Data mining tasks are applied based on the datasets containing various patterns. Classification, association rule mining, clustering and Bayesian networks are few of the data mining tasks [36]. A well designed privacy preservation technique should support multiple data mining tasks, while maintaining data quality. However, data quality is maintained by only a group of data mining tasks [9].

Data distribution refers to the division of data. Data sets can be either centralized or distributed. A centralized data set is owned by a single party. On the contrary, distributed data sets are divided between two or more parties, who most probably do not trust each other but are interested to perform data mining techniques on their unified data. Distributed data can be classified as horizontally and vertically partitioned. In horizontally partitioned data each party has the same set of attributes but different set of records. In vertically partitioned data each party has the same set of records but different set of attributes.

The dimensions proposed in [9] are similar to [32] but they add one more extra dimension, the data types. There are two basic data types: numerical and categorical (nominal). Boolean data are a special case of nominal data. The basic difference between the two data types is that categorical data are categorized without a natural rank, but on the other hand numerical data are instantly measured by a number. This difference creates the need to take different privacy preservation approaches.

Privacy is defined differently based on the context [9]. Either data values are sensitive and need to be protected or certain association or classification rules are private. In the first case, privacy preservation focuses on individual values such as personal identification information which can be linked to a specific individual. Thus the information related to an individual must remain private and be protected from disclosure [1]. In the second case, privacy preservation focuses on protecting from disclosure of sensitive information related to a group. Therefore, the privacy preserving techniques depend on how privacy is defined [1, 9].

The most important dimension is the privacy preservation technique that is used for the protection of data such as data modification and secure multi-party computation (SMC). Data modification methods perform modification on the original values of a database, before releasing to the public, minimizing information loss. Data modification methods are data perturbation, data swapping, aggregation and suppression. Perturbation is accomplished by altering an attribute value or adding noise in numerical attribute values drawn from a normal distribution with zero mean or standard deviation [9]. Data swapping is used in nominal values and replaces original values in order to create a non identified record, focusing on preserving patterns instead of statistical properties. Values of individual records are interchanged and in

the sampling method only a sample of data is released. Aggregation is used for protecting privacy by perturbing the original data set before releasing. In suppression, sensitive data are deleted or suppressed before releasing.

Secure multi-party computation, a cryptographic-based technique introduced by Yao [34], allows secure computation and privacy is preserved if at the end of the computation none of the parties involved know anything other than their own input and the final results [2, 34]. The datasets are encrypted but data mining operations can still be applied. This technique is based on cryptographic protocols and applied to distributed datasets. The basic idea is that the parties involved in the data mining process, encrypt their data and send it to other parties. More details on this technique can be found in Section 2.2.5.

2.2 Privacy preserving techniques categorization

Over the years, privacy preserving data mining has been studied extensively by the data mining community and many techniques have been proposed. Different privacy preserving data mining techniques can be categorized into the following categories: anonymization, perturbation, randomization, condensation, and cryptography [5, 23, 32]. In the following subsections these methods are further presented.

2.2.1 Anonymization technique

Anonymization refers to the method in which an identity or/and sensitive data about a record or individual need to be protected. The main goal of anonymization methods is to make each individual/record indistinguishable among the other individuals/records. Techniques such as generalization and modification are used to achieve this goal [37–39].

Data in a table can have four types of attributes [5]: explicit identifiers, quasi identifiers, sensitive attributes and non-sensitive attributes. The name or the tax number are examples of explicit identifiers which can identify an individual explicitly. Many attributes can be considered quasi-identifiers which can be used and combined with public data in order to uniquely identify records and/or individuals. The salary, the disease, etc are few of the attributes that are considered sensitive as they consist of personal information about individuals. Attributes that can be revealed and there is no need to be protected constitute the non-sensitive attributes.

The most representative anonymization approach is the k-anonymity. Based on this approach any individual/record is k-anonymous if each individual/record is indistinguishable from at least $k-1$ other individuals/records. It is obvious that explicit identifiers should be removed but privacy can still be violated if quasi-identifiers are combined with public data. This created the need to protect privacy and develop well-designed anonymization techniques, such as k-anonymity, and other methods beyond k-anonymity such as l-diversity, t-closeness etc. K-anonymity is extensively described in Section 6.2 and Section 6.2.1.

2.2.2 Perturbation technique

Perturbation methods distort data prior to data mining. These methods replace original values with different ones such that the statistical information computed

from both the original and perturbed data is the same. Perturbation techniques aim to hide the individuals represented by the original data, by performing synthetic changes in the original records so that statistical properties are preserved. This can be achieved by adding noise or data swapping. Perturbation methods do not reconstruct the original values, so for each data mining approach, i.e classification, clustering or association rules, need different methods. Perturbation treats attributes independently, thus an important disadvantage of this method is the loss of information.

2.2.3 Randomization technique

Randomization is a perturbation technique where data are masked by random data. Randomization methods aim to find a balance between privacy preservation and knowledge discovery [5]. These methods add noise to data in order to mask the original values and protect privacy in data mining operations. The noise that is added is enough so that original values cannot be recovered. By recovering the probability distribution of the aggregate data, it can be used for privacy preservation purposes. Decision tree classification is based on aggregate values so this method is very useful. Randomization is carried out in two main steps: firstly the data are randomized and secondly they are transmitted to data receivers who reconstruct the original distribution of the data by employing a reconstruction algorithm. This is a simple method and does not require information about the distribution of the other records [1]. Thus, randomization can be implemented at the data collection phase and the existence of a trusted third party is not required. Since the knowledge of the distribution of all records is not required, randomization treats all the records equally disregarding their local density. A solution would be to add noise to all the records, but this results in the reduction of the data utility.

2.2.4 Condensation technique

Condensation approach is using condensed statistics of the clusters in the dataset to generate pseudodata. Groups of non homogeneous size are constructed such that each record belongs to a group whose size is at least equal to its anonymity level. Each group generates pseudodata in order to create a data set that has the same aggregate distribution with the original dataset which can be used by a variety of data mining problems. Privacy is better preserved by this methodology as data are not modified, and pseudodata is used which has the same format as the original data. However, the data mining results are largely affected as information is lost due to the condensation of the records into groups [5].

2.2.5 Cryptography-based technique

Database owners wish to conduct data mining operations jointly with other datasets distributed in different locations. For example, multiple institutions with medical data wish to conduct research which will benefit all parties involved, but sensitive information should remain secret. These operations can take place between mutually untrusted parties, and for that reason the basic task in distributed privacy preserving data mining is to preserve the privacy of the inputs the owners are providing but also

their identities. Cryptography is ideal for these scenarios, where untrusted parties collaborate in order to extract useful information but preserve their privacy at the same time while utility remains in high levels. Data can be horizontally or vertically distributed. This method reveals only the final results, and nothing more.

Most proposed methods in the literature based on cryptography follow the same encryption protocol known as Secure Multiparty Computation (SMC). There are two main adversarial models: the semi-honest and malicious. In the semi-honest model, each party involved follows the protocol, but may be curious to learn the sensitive information of the other parties. In the malicious model, the adversarial party deviates from the protocol, and tries to learn the private information of the participants to the protocol.

There are two types of distributed privacy-preserving data mining protocols [12]: a set of secure protocols that prevent information disclosure and a set of operations used in data mining algorithms and suitable for preserving privacy. In the first type, the oblivious transfer protocol and the homomorphic encryption are included. The oblivious transfer protocol is by definition a two-party protocol. In the homomorphic encryption concept the objective is to perform algebraic operations on an encrypted message such that the decrypted result is the same as the result of the algebraic operation on the plain message. In the second type, the operations that are used in both data partitions are: the secure sum, the secure set union, the secure size of intersection and the scalar product [40]. The secure sum allows to obtain the sum of the inputs from each party without revealing the inputs to the other parties. The secure set union creates unions of sets without revealing the set owners. The secure size of set intersection anonymizes the owner of the data by computing the size of the intersection of the local sets. The secure scalar product uses random values to an input and the final result is obtained by removing the randomness. In these types, encryption techniques are used to prevent information disclosure.

The results of this approach are secure and exact but in case many parties are involved this method may not be efficient. Also this method does not guarantee that the disclosure of the final results may not pose threat to the privacy of individual records.

2.2.6 Hybrid technique

Many techniques have been proposed in the privacy preservation field in order to protect the data. However, there is no single technique that is consistent in all domains. Each technique has some advantages but also some limitations and each one performs in a different way based for example on the type of data or the application. Because of the limitations of each method, two or more techniques can be combined in order to overcome the privacy issues each method may pose. This approach of merging more techniques is called hybrid technique.

For example, randomization and generalization can be combined, by applying randomization on the original data and then the modified data are generalized, providing results with better accuracy. Another example is the combination of perturbation and generalization, reconstructing the original data in order to provide results with no information loss. The authors in [27] combined noise addition with cryptographic techniques for secure mining of association rules.

2.3 Advantages and disadvantages of PPDM techniques

Different privacy preserving data mining methods result in different advantages but also disadvantages. Based on the technique, privacy can be preserved but there might be information loss, or the complexity might result in an non-applicable method. Table 2.1 presents the benefits and drawbacks of the different privacy preserving data mining techniques [41, 42].

Table 2.1: Advantages and disadvantages of PPDM methods.

Method	Advantages	Disadvantages
Anonymization	Generalization or modification. Prevents identity disclosure while releasing sensitive information.	Results in information loss to some extent. Vulnerable to linking attacks. Prone to homogeneity attacks and background knowledge attacks.
Perturbation	Simple technique. Adding noise. Independent treatment of distinct attributes.	Distortion is the only way to reconstruct the original value. Ambiguity in degree of equivalence of different records.
Randomization	Simple technique. Adding noise. More efficient. Easily implemented. No need for a server. Useful for hiding individual sensitive data. Can be implemented at data collection phase.	Treats all the records equally and reduces the utility of the data. Not appropriate for several attribute databases. Information loss.
Condensation	Aggregation. Suitable for pseudo-data. Better approach than modification in original data.	Pseudo-data has the same format as the original data.
Cryptography	Well suitable approach. Provide protection of sensitive information.	Scaling is difficult when more parties are involved.

2.4 Knowledge protection

Most methodologies focus on preserving the data privacy, however, some methods explore the protection of sensitive knowledge patterns that can be revealed after the data mining processes. These methods modify as well the original dataset, but in a way that will secure the disclosure of certain sensitive knowledge patterns. Methodologies have been proposed for hiding sensitive knowledge in the context of association and classification rule mining.

Association rule mining has as its main goal to produce a set of interesting and useful rules [43]. A rule in a dataset is quantified based on the confidence and support. The association rules whose confidence and support are above a specified threshold are mined, but these rules may be sensitive according to the data owner. All sensitive association rules should be hidden in the sanitized dataset and non-sensitive rules should be available and not sanitized, however there is a cost on the utility. The sanitization process has to be accomplished so that the general patterns of the dataset are preserved, and sensitive knowledge is secret. Heuristic approaches related to association rule hiding have been studied by the majority of researchers due to their efficiency and scalability [44].

In classification rule hiding a set of classification rules is considered sensitive and suppression or reconstruction techniques are used to protect from their disclosure. In suppression techniques, the confidence of a classification rule is reduced by distorting a set of attributes in the dataset that belong to transactions related to its existence. In reconstruction approaches, the dataset is reconstructed by using only transactions which support non-sensitive classification rules, thereby leaving the sensitive rules unsupported [44]. Classification rule mining has been studied less compared to association rule mining.

2.5 Evaluation criteria

An important characteristic in the development of PPDM algorithms is the recognition of appropriate evaluation criteria. The already-developed privacy preserving algorithms do not outperform all other algorithms on all evaluation criteria. An algorithm may perform better than another one for specific criteria [32]. As such, different sets of metrics for evaluating these algorithms have been proposed over the past years. Quantifying privacy is challenging. Many metrics have been proposed in the literature; however, multiple parameters need to be evaluated. Most of the proposed metrics can be classified into three main categories depending on the aspect being measured:

1. Privacy level metrics: the security of the data from a disclosure point of view;
2. Data quality metrics: quantify the loss of information/utility;
3. Complexity metrics: measure the efficiency and scalability of the different techniques.

Both data quality and privacy level can be further categorized as data metrics and result metrics. Data metrics evaluate the privacy level and data quality by estimating the transformed data resulting from applying a privacy preserving

methodology. Result metrics evaluate the privacy level and data quality by estimating the outcomes of the data mining process having the transformed data as the input [12].

Verykios et al. [32] provided a different list of evaluation criteria to be used for assessing the quality of PPDM algorithms:

- the performance of the algorithm in terms of time needs to hide sensitive information,
- the data utility after the PPDM technique is applied, which is equivalent to the minimization of information loss,
- the level of uncertainty with which the sensitive information hidden can still be discovered and
- the resistance accomplished by the privacy algorithm to different data mining techniques.

Sharma et al. [9] proposed a different set of evaluation criteria:

- Versatility: the ability of a technique to serve various data mining tasks, privacy requirements, and data set types. The technique is more useful if it is more versatile.
- Disclosure risks: the possibility that a malicious party obtains sensitive data. Preservation techniques aim to minimize the risks.
- Information loss: the decrease in data quality resulting from the noise added to the data and the level of security applied. A privacy preserving technique is required to maintain the quality of data in the released data sets. If data quality is not maintained, the use of security is purposeless.
- Cost: the computation and communication costs. The computational cost depends on the processes applied on the data, for example, randomizing the database values, and the cost to run all processes. The higher the cost, the more inefficient the technique.

Different parameters were also defined [23, 45] to quantify the trade-off between privacy and utility. The authors created a framework for evaluating PPDM algorithms, indicating the importance of designing adequate metrics that can reflect the algorithm properties and developing benchmark databases to test and evaluate all types of algorithms. They identified a framework based on the following dimensions to evaluate the effectiveness of PPDM algorithms:

- ☐ Efficiency is the ability of a privacy-preserving algorithm to execute with good performance.
- ☐ Scalability evaluates the efficiency of a PPDM algorithm with increasing data set sizes.
- ☐ Data quality is the quality of both the input data and the final data mining results.

- ☐ Hiding failure is the portion of sensitive data that is not hidden after the PPDM technique is applied.
- ☐ The privacy level, which results from the use of a privacy preserving technique, indicates how closely the sensitive information can still be estimated.
- ☐ Complexity refers to the execution of an algorithm in terms of performance.

Qi and Zong [46] described evaluation criteria and reviewed privacy protection algorithms in data mining such as distortion, encryption, privacy, and anonymity. Malik et al. [5] also presented evaluation parameters and discussed the trade-off between privacy and utility. They suggested that practical algorithms need to be developed that balance disclosure, utility, and costs to be accepted by industry. They stated that novel solutions have been developed but product-oriented solutions need to be developed so that real-world problems are efficiently handled.

2.6 Data quality

One of the most important properties of data is their quality. Data often are sold or shared for research purposes and should have a certain level of quality based on their potential usage. If the quality of data is high the more useful is the information of the data contained within the database. Operations, such as perturbation techniques, applied to sanitize sensitive information downgrade the data quality, which may result in economical or social damages, or become useless for the purpose of knowledge research. Data quality needs to be taken into account for the evaluation of a privacy preserving data mining technique [23].

Bertino et al. [23] try to identify a set of possible measures that can be used for the evaluation of data quality, after privacy preservation techniques have been applied. After privacy preserving processes take place, the evaluation of the data quality can be useful to qualify both the data and the data mining results. The authors consider three main parameters:

- Accuracy: measures the proximity of a sanitized value to the original.
- Completeness: evaluates the degree of missed data in the sanitized database.
- Consistency: is related to the relationships among different fields of a data record or among data records in a database.

Accuracy is a general parameter that can be measured on the analyzed data, on the other hand completeness requires to determine all the relationships that are relevant for a given dataset. Accuracy as a measure of the quality of data is closely related to the information loss. Measuring accuracy depends on the specific privacy preserving data mining algorithms. For example, if the algorithm performs perturbations, the information loss can be measured by measuring the dissimilarity between the original and sanitized dataset. If the algorithm is using data swapping, the information loss can be measured by a parameter measuring the data confusion introduced by the value swappings. On the other hand, cryptography-based algorithms do not use any perturbation technique in order to preserve privacy. These methods assure data privacy through cryptographic techniques, which guarantee that the quality of data is not compromised.

2.7 Proposal

The proposed framework exploits encryption mechanisms such that privacy is preserved but at the same time useful information can be extracted through data mining operations. The cryptography-based approach, as discussed, assures that the data remain private but also guarantees that the data quality is not affected.

The protocol which is discussed thoroughly in Section 4, is designed for a distributed environment. In particular, a miner who acts as the data collector is connected with at least three parties, each one communicating with the miner in order to send their data and create the data mining model. The proposed privacy preserving data mining approach was firstly designed only for horizontally partitioned databases [47] and later extended to support vertically partitioned databases [48,49] exploiting the multi-candidate election schema [50] aiming to extract global information from both partition types. The approach handles both nominal and numeric database values.

Traditional Naive Bayes classifier is widely used in the literature for privacy preservation techniques, but is based on the unrealistic assumption that attributes are independent. On the contrary, the current proposal utilizes the Tree Augmented Naive Bayesian (TAN) classifier [51] which eliminates this assumption and behaves more robustly. The privacy preserving version of this classifier was properly designed and developed for the purposes of the proposed implementation. The Paillier cryptosystem [52] was implemented to perform all necessary cryptographic processes to preserve privacy, by exploiting the homomorphic primitive, first proposed by Yang et al. [53]. Based on this primitive, the data collector (miner) and each participant are unable to identify the original data of the shared distributed databases, except naturally the data owner. In addition, the identity of the database owners is private and indefinable by any aggressor. Communication among participants is unfeasible, and the miner is able to continue with the performance of all necessary operations, if at least three participants are connected with the data collector. Integration mechanisms assure that all data transmitted are not modified as a summary is concatenated to each message by applying the SHA-1 hash function. The only data revealed is the final results to each one participant.

The protocol is safe from various types of attacks which are presented in Section 4.4. From the evaluation, the approach is considered efficient and there is a balance between privacy preservation and knowledge discovery. Specifically, the protocol is divided in six main phases:

Phase 1. Key generation All participants and the data collector create their encryption key pairs, and a 1024 digital signature.

Phase 2. Mutual authentication Each party and the miner are mutually authenticated using their digital signatures. We assume all parties are able to acquire the public keys of each other participant.

Phase 3. Data collection The miner collects the data of each one of the participants encrypted. In particular, requires only the frequencies of each attribute value related to each class value. The homomorphic primitive is applied to ensure that privacy is not violated.

Phase 4. Classifier initialization After the data collection, the classifier is successfully initialized when at least 3 parties have sent their data and participate in the creation of the mining model.

Phase 5. TAN classifier creation If all above phases are complete, the miner is in position to create the TAN classifier and create the final mining model.

Phase 6. Final results As the creation of the TAN classifier is complete, the miner sends the final results encrypted to each participant.

2.8 Contribution

Privacy preservation has gained a lot of attention in the data mining community. Many studies were presented related to this field. However most of the techniques proposed in the literature are theoretical or empirical. The authors present solutions that lack implementation of the hypothesis presented in their work.

Methods that have been proposed for privacy preservation handle either horizontally or vertically partitioned databases. None of the privacy preserving algorithms in the literature handle both partition types. In addition, these approaches mainly focus on nominal attribute values. To our knowledge, in the privacy preserving research field, the implementation of algorithms that support both horizontally and vertically partitioned databases have never been proposed in the past. On the contrary, the current presented privacy preservation protocol is implemented to support both partition types. The developed system can handle both nominal and numeric attribute values, including binary values.

The cryptographic-based approach can assure that data remain secret and at the same time the utility of data is assured. As presented in Section 4.3, the proposed implemented protocol can preserve privacy and can confront a number of well known attacks on distributed systems.

Zhang et al. [54] presented a technique similar to our approach. They apply the Tree-Augmented Naive Bayes mining technique, the same as the mining method in the current approach, however their proposed method handles only horizontally partitioned databases. Moreover, they apply their method only on numeric attribute values as they exploit perturbation mechanisms for preserving privacy. Perturbation approach however can result in the decrease of data utility, which is avoided in case cryptography is applied. Therefore, the method proposed in Chapter 4 is considered a better approach as related to the accuracy of data while data mining operations are applied in a distributed environment while preserving privacy.

2.9 Organization of Part I

The structure of Part I is as follows. Chapter 3 summarizes privacy preserving data mining methods proposed in the literature. The proposed privacy preservation protocol is presented in Chapter 4. The background of the current approach is defined in Section 4.1. Section 4.2 describes the proposed protocol and its security and design requirements. The evaluation of the current protocol in terms of performance and data accuracy is presented in Section 4.3 while Section 4.4 analyzes some possible threats to the current proposal and how they are confronted. A brief summary of Part I is given in Chapter 5.

Chapter 3

Related work

Privacy preserving data mining received attention and widely researched through the recent years and became an important topic in data mining research, since the work presented in [18] and [19]. Privacy preserving data mining techniques have several applications on different domains. Some of the domains raise concerns about the disclosure of sensitive information.

The majority of privacy preserving data mining techniques developed to prevent leakage of sensitive information, without undermining the extracted knowledge produced by the application of mining processes on data [44]. The methods applied either modify or remove some original data to achieve privacy preservation. This action creates a trade-off between the data quality and the privacy level, known as utility. Privacy preserving data mining techniques should be designed in order to guarantee the maximum utility of the produced outcomes while an appropriate level of privacy is achieved.

Common approaches of privacy preservation in data mining are data distribution, data distortion, data hiding, rule hiding, k-anonymity, randomization, etc [55]. A simplified categorization of these approaches is given in [5]. Common goal of all these methods is to provide effective results while reaching a trade-off between the privacy level and the data mining technique performance [5].

Existing privacy preserving data mining methodologies [44] can be divided into methodologies that protect the input data in the mining process, and methodologies that protect the final results of the mining process. Privacy preservation techniques (perturbation, generalization, transformation, etc), in the first methodology, are applied to the input data to hide any private information and distribute the data to other parties with safety. The main goal is the generation of accurate data mining results. SMC methods enable data owners to apply mining methodologies on their data, keeping the datasets secret. In the second approach, the applied privacy preservation techniques prohibit the disclosure of private information derived through the application of data mining algorithms.

Verykios et al. [32], categorize privacy preserving data mining algorithms in five segments. The first segment is the data distribution, and refers to the division of data, either centralized or distributed. Data modification, the second segment, is used in order to modify the original database values. The databases may need to be released to the public, so modification ensures the protection of privacy. Data mining algorithm, the third segment, is the algorithm for which the data modification is taking place and for which the privacy preservation technique is designed.

The most important algorithms have been developed for classification, like decision trees, association rule mining algorithms, clustering algorithms and Bayesian networks. Data or rule hiding, the fourth segment, refers to whether sensitive values should be protected by hidding raw or aggregated data. The complexity for hiding aggregated data is higher, and for this reason, mostly heuristics have been developed. In some cases individual data values are private, but in other cases individual association or classification rules are considered private. Depending on how privacy is defined, different privacy preserving techniques are applied. The most important fragment is the privacy preservation technique. These techniques can be categorized to heuristic-based, reconstruction-based and cryptography-based techniques. Heuristic techniques modify selected values rather than all available values, in order to minimize the information loss. In reconstruction techniques, the original distribution of the data is reconstructed from the randomized data. Though, data modification results in degradation of the database performance. In cryptographic techniques, i.e. SMC, a computation is secure if at the end of it, no one knows anything except its own input and the final results. These methods are considered for preserving privacy in distributed environments by using encryption techniques.

As defined by the authors [56], every privacy preserving methodology should answer one major question: "*Do the results themselves violate privacy?*". In other words, do the results of a data mining process violate privacy by exposing sensitive data and patterns that can be used by attackers? A privacy preservation classification model is proposed by the authors, and they study possible ways an attacker can use the classifier and compromise privacy, but they do not provide a solution to prevent an attacker from accessing the mining results and thus violate privacy.

Scardapane et al. [57] analyze distributed medical data in multiple parties. Medical environments may forbid, due to privacy restrictions, to disclose their locally produced data to a central location.

Sweeney [31, 58] proposes a heuristic approach using generalization and suppression techniques to protect raw data and achieve k-anonymity. A database is k-anonymous, with respect to some attributes, if at least k transactions exist in the database for each combination of the attribute values. The new generated database guarantees the k-anonymity by performing generalizations on the values of the target attributes. Zhong et al. [59] used a third-party to achieve k-anonymity in horizontally partitioned databases.

More details on the k-anonymization approach is given in Section 6.2.

3.1 Perturbation-based techniques

The most widely studied privacy preservation techniques are cryptography and randomization. Agrawal et al. [43] presented a framework for preserving privacy by randomizing nominal values for mining association rules. A naive Bayes learning technique was applied in [60] to construct differentially private protocols to extract knowledge from distributed data. A multiplicative perturbations approach was applied on the data for introducing noise by Liu et al. [61]. However, perturbation techniques decrease the quality of the final results. Also, the authors in their privacy analysis did not consider any prior knowledge.

Vaidya et al. [62] apply differential privacy to develop a naive Bayes classifier provided as a cloud service and focus on generating privacy preserving results instead of sharing secure data sets. These techniques mainly focus on publishing useful results and not sanitized data that can be shared.

Randomization techniques were used in the past to build association rules [63] and decision trees [18] for vertically and horizontally partitioned databases respectively. Du and Zhan [64] also proposed a method for building privacy preserving decision trees. Evfimievski et al. [65] proposed privacy preserving association rule mining based on randomization techniques and guaranteed privacy. Vaidya and Clifton [63] studied association rule mining and proposed an algorithm based on the Apriori algorithm to extract the candidate set for vertically partitioned data.

The randomization method even though is efficient, can result in inaccurate outcomes. As revealed by the authors in [66], randomization techniques may compromise privacy. The authors point out that additive noise can be easily filtered out, and special attacks can result in the reconstruction of the original data. A randomization technique that combines data transformation and data hiding was proposed by Zhang et al [67]. They exploit a modified naive Bayes classifier to predict the class values on the distorted data.

Agrawal and Srikant [18] build a decision tree classifier from applying perturbation techniques on the training data and estimate the distribution probability of numeric values. They propose a measure and evaluate the privacy offered by their method. The privacy is measured by how closely the original values can be determined through the modified data.

The approach presented in [33] is another reconstruction technique based on an Expectation Maximization algorithm for distribution reconstruction. The authors provide metrics for quantification and privacy and information loss measurement. Unlike the approach in [18], the metric proposed in [33] takes into account the fact that the perturbing distribution as well as both the perturbed record and the reconstructed distribution are available to the user.

3.2 Cryptography-based techniques

On the contrary, cryptographic-based techniques are more secure. They provide accurate results but they lack efficiency. Most cryptographic methods proposed in the literature are based on the idea of Yao [34], and an extension proposed by Goldreich [2], who studied the secure multi-party computation problem.

A few proposed privacy preservation techniques apply encryption mechanisms on horizontally partitioned databases for building decision trees [19, 68]. A variety of cryptography based techniques are applied on naive Bayesian classifiers [28, 53, 69, 70].

Kantarcioglu and Clifton [27] applied cryptography to build association discovery rules over horizontally partitioned data. Yang et al. [53] focus on horizontally partitioned data where each party has access to its own record. Tassa [71] focuses on horizontally partitioned databases and proposed a protocol for secure mining of association rules, presenting the protocol's advantages over existing protocols [27].

On the other hand, the authors in [29, 40] and [69, 72], benefit from the cryptographic methods and apply them on vertically partitioned databases to create association rules and naive Bayesian classifiers, respectively. Vaidya and Clifton [73] focus on vertically partitioned data and proposed a privacy preserving k-means clustering algorithm, where clusters are based on their similarity. Du and Zhan [74] consider two parties to construct ID3 on vertically partitioned databases. Fang et al. [75] created a decision tree model for horizontally partitioned data based on homomorphism encryption. Fang et al. [76] proposed a decision tree classification for vertically partitioned data. Vaidya and Clifton in [77] proposed a clustering approach over vertically partitioned data.

Goethals et al. [78] proposed a simple and secure method, applying secure multiplications. Similarly, in [79], the authors propose a multi party approach to calculate the aggregate class for vertically partitioned data applying Naive Bayes classifier. Because of its simplicity and straightforward method, Naive Bayes classification is utilized by many researches [28, 53, 70, 80].

Yu et al. [81] propose a method over vertically partitioned data for privacy preserving SVM classification, computing the global SVM model without revealing data or classification information to other parties. Jiang and Clifton [82] use exchange encryption to completely anonymize vertically distributed data and hide sensitive information in the communication process.

Other data mining methods have been proposed in the privacy preserving data mining field, such as tree augmented naive Bayes [54] and the K2 algorithm [29].

Kumbhar and Kharat [83] proposed an algorithm based on homomorphic encryption, secure scalar product and Shamir's secret sharing technique for vertically partitioned databases used for association rule mining. The authors also proposed an algorithm for horizontally partitioned databases based on homomorphic encryption with a combination of RSA public key cryptosystem.

Zhang et al. [54] proposed a similar approach to the current proposed methodology. However, they apply an algebraic technique to perturb the original data. Instead, our protocol exploits cryptographic-based techniques, assuring privacy and resulting in more accurate outcomes.

In Appendix B, a comparison of some privacy preserving data mining techniques proposed in the literature are presented in Table B.1.

Chapter 4

Privacy preservation framework

4.1 Background

In machine learning and statistics, classification refers to a supervised predictive learning approach where a class value is predicted from data given as input. In its simplest form is the ordering of data into groups based on their similarities. The difference between clustering and classification is that classification uses predefined classes, while clustering is used to establish such classes/groups. Classification can be performed on both structured or unstructured data. The main goal of the approach is to identify the class of new data. Classification algorithms require as input training data to predict the likelihood that future data will fall into one of the predetermined classes. The learning model is trained using the training data and the performance is measured using test data. Common classification problems are speech recognition, face detection, handwriting recognition, document classification, credit approval, medical diagnosis, target marketing etc.

4.1.1 Classification of nominal attributes

The main objective of classification is the prediction of an attribute value given a training set by estimating the probabilities. Given an attribute X with nominal values $x_1, \ldots, x_r$, the calculation of the probability of each value is given by applying Equation (4.1), where n is the total number of training instances for which $V = u_j$ and n_j is the number of instances that have $X = x_k$.

$$P(X = x_k|u_j) = n_j/n \tag{4.1}$$

The conditional probability that an instance belongs to a certain class c is calculated by Equation (4.2), where n_{ac} is the number of instances with class value c and attribute value a, and n_a is the number of instances with attribute value a.

$$P(C = c|A = a) = \frac{P(C = c \cap A = a)}{P(A = a)} = \frac{n_{ac}}{n_a} \tag{4.2}$$

4.1.2 Classification of numeric attributes

The calculations of the classification probabilities differ for numeric and nominal attributes. The mean μ and variance σ^2 parameters, for numeric attributes, are calculated for each class and each attribute. The probability $P(X = x'|u_j)$ that an instance is class u_j can be estimated by substituting $x = x'$ in the probability density equation. The conditional probability of a class is calculated for all classes, and the class with the highest relative probability is chosen as the class of the instance. These local sums are added together and divided by the total number of instances having that same class to compute the mean μ for a class value. Each party, since it is aware of the class of the training instances, can subtract the appropriate mean μ from an instance having class value y, square the value, and sum all such values together. The required variance is obtained by dividing the global sum by the global number of instances having the same class y.

Equation (4.3) computes the normal probability distribution, where x is a random variable, μ is the mean of the distribution and σ is the standard deviation (σ^2 is the variance), π is approximately 3.14159 and e is approximately 2.71828.

$$P(x) = \frac{1}{\sigma * sqrt(2\pi)} * e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{4.3}$$

4.1.3 Tree augmented naive Bayesian classifier

The traditional naive Bayes classification (Figure 4.1) is a method based on Bayes theorem. Naive Bayes classifiers are simple, easy to build, and useful for very large data sets as they are highly scalable. Naive Bayes classifiers support both nominal and numeric attribute values. These classifiers compute the conditional probability of each attribute value A_i given the class value C. The Bayes theorem is applied to compute the probability of class C given a specific instance vector $< A_1.....A_n >$, given the total number of n attributes.

These classifiers assume that all attributes are conditionally independent given the value of C, which is a restrictive and oversimplified assumption, reducing the computational cost by only counting the class distribution. However, in most cases, this assumption is unrealistic, as some attributes can be dependent. Since prior knowledge of the class variable C is not considered, a bias in the estimated probabilities is introduced, which leads to poor prediction outcomes in some domains [84]. The performance of such classifiers can be improved by removing this assumption.

One method to reduce the naive Bayes' bias is to relax the independence assumption using a more complex graph. An interesting variation of Bayesian networks is the tree augmented naive Bayesian (TAN) classifier (Figure 4.2) [51]. TAN can be viewed as a Bayesian network, a probabilistic graphical model, where each attribute has the class as the parent, and possibly an attribute as a second parent. The existence of additional edges between attributes, which represent the correlation among these attributes, is allowed by the TAN classifier. More specifically, in a TAN network, the class C has no parents and each attribute A_i has the class and at most

one other attribute A_j as parents, implying that the assessment of the class of attribute A_i also depends on the value of A_j. For example, in a dataset, the age of an individual and their financial income are two dependent attributes.

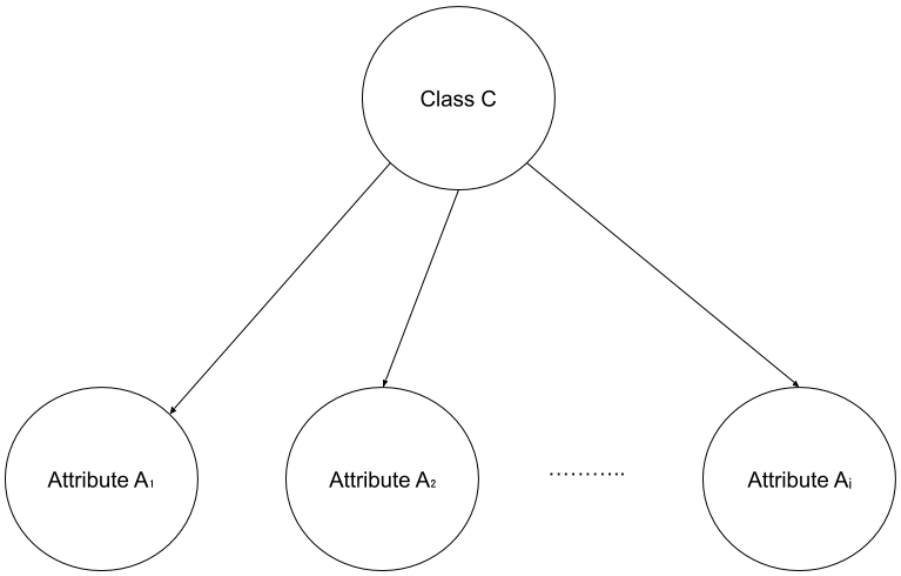


Figure 4.1: Bayesian network structure.

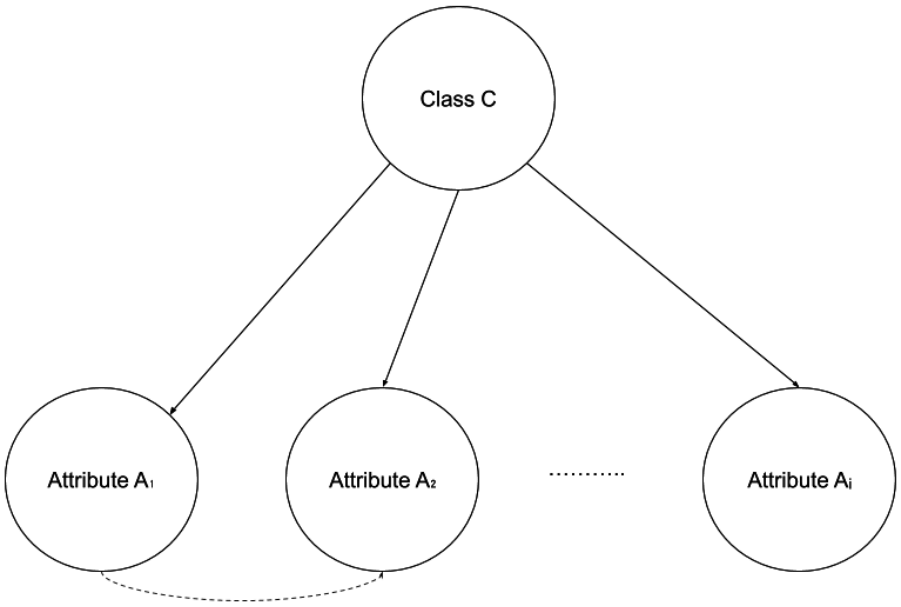


Figure 4.2: TAN structure.

The procedure of learning these edges, which is based on a method proposed by Chow and Liu [85], reduces the problem of constructing a maximum likelihood tree to find a maximal weighted spanning tree in a graph. The problem of finding such a tree involves selecting a subset of edges such that the sum of weights attached to the selected edges is maximized. The TAN algorithm consists of four main steps:

1. The mutual information for each attribute pair is computed using Equation (4.4), measuring how much information the attribute y provides about x.

2. An undirected graph is built in which the vertices are the variables in x (the weight of an edge connecting two attributes).

3. A maximum weighted spanning tree is created.

4. The undirected tree is transformed to a directed one by choosing a root variable and setting the direction of all edges to be outward from it.

$$I_p(X;Y) = \sum_{x,y} P(x,y) log \frac{P(x,y)}{P(x)P(y)} \tag{4.4}$$

The TAN classifier, by removing any independence assumptions, behaves more robustly with regards to classification compared to the classical naive Bayes classifier, since it combines the initial structure of the naive Bayes algorithm with prior knowledge (if available) or obtained knowledge about the correlation of input attributes via a training approach. TAN substantially reduces the zero-one loss of naive Bayes on many data sets and a range of experiments have shown that it outperforms the naive Bayes classifier [51, 86]. TAN results are significantly improved compared to those produced by the classical naive Bayes classifier and Bayesian networks. The robustness and computational complexity are also maintained, showing better accuracy.

4.1.4 Homomorphic primitive

Homomorphic encryption is widely used in the literature [78,80,87,88] for approaches implementing cryptography-based techniques. The homomorphic primitive was first used to build a privacy-preserving data mining model in a distributed environment by Yang et al. [53].

$$E(M1 \otimes M2) = E(M1) \otimes E(M2) \tag{4.5}$$

This primitive allows the performance of calculations on encrypted data without the need to decrypt these data. Equation (4.5) describes the operation where the result of encrypting two messages is equal to the sum of the two messages separately encrypted.

4.1.5 Paillier cryptosystem

The additive homomorphic primitive is exploited by the Paillier algorithm [52]. Through this primitive, anonymity and unlinkability between parties and personal data are achieved [69].

During the key generation phase of the Paillier cryptosystem, each participant (the miner and all parties involved) generates a key pair of 1024 bits size on their own side. The public key of each party is the product N of two random prime numbers ($N = p * q$), which are independent and have the same size, and a random number g, which belongs to $Z^*_{n^2}$. The private key is the result of variables *lambda* shown in Equation (4.6) and mu, defined in Equation (4.7).

$$Lambda = lcm(p-1, q-1) = (p-1) * (q-1)/gcd(p-1, q-1) \tag{4.6}$$

$$mu = (L(g^{lambda} \mod N^2)^{-1} \mod N), \quad \text{where } L(u) = (u-1)/N \tag{4.7}$$

Paillier encryption is performed as shown in Equation (4.8). In the proposed protocol, more specifically, if a participant j is interested in participating in forwarding the frequency i to the miner, then the party needs to encrypt every message sent with the miner's public key. The cryptosystem is vulnerable to chosen-plaintext attacks. For confronting these types of attacks, a random variable M is computed by the miner, and delivered to each party encrypted with their own public key. The M variable is used for encrypting every transmitted message.

The current approach requires the participation of at least three parties. When all three parties have forwarded their data to the miner, the homomorphic primitive is applied. The miner calculates the total frequencies of each possible attribute value in relation to each class value by decrypting all the received messages simultaneously. The miner is not in a position to associate the received frequencies with the original records and cannot link the data to their owners due to the execution of the decryption process after the participation of at minimum three parties. A decrypted message is presented in Equation (4.9).

$$E[m_{i,j}] = g^{M^i} x^N (\mod N^2) \tag{4.8}$$

$$T = a_0 M^0 + a_1 M^1 + ... + a_{l-1} M^{l-1} (\mod N) \tag{4.9}$$

4.2 Privacy preservation protocol

Challenges arise during the execution of data mining processes when preserving privacy, since the collected data being mined often contain sensitive information. Data mining techniques used to derive statistics from distributed databases should ensure that personal data will not be disclosed to unauthorized individuals. The objective of the proposed framework is to develop a privacy-preserving protocol that satisfies the essential security and design requirements, exploiting efficient encryption mechanisms. The tree augmented naive Bayesian classification algorithm [85] is used to extract accurate and global information while preserving privacy.

Encryption processes are applied to a client–server (party–miner) environment ensuring that any message exchanged in a fully distributed environment is not accessible by internal or external attackers, either by the parties involved or the miner. The miner generates the classification model by collecting the frequencies of each attribute value in relation to each class value from at least three horizontally or vertically partitioned databases, which are owned by different parties. In vertically partitioned databases, we assume that every participant is aware of the class value of each record. The proposed protocol was developed for supporting both nominal attribute values (Algorithm 1) and numeric attribute values, including binary data (Algorithm 2). Through the Paillier cryptosystem, all frequencies forwarded are encrypted. The exploitation of the homomorphic primitive ensures that sensitive data remain protected. Communication among parties is prohibited and the only data flow occurs between each party and the miner, making communication among parties infeasible.

As mentioned, the current work is an extension of previous research [47–49]. Notably, some of the features and requirements used arise from the quotations presented by Mangos et al. [87].

4.2.1 Design and security requirements

Each developed protocol must implement appropriate measures and follow data protection principles to safeguard individual rights, as defined by the General Data Protection Regulation (GDPR) [7]. Privacy and data protection must be considered at the design phase and throughout the entire life cycle of any protocol and system, as defined by the Privacy by Design approach. The development and implementation of the current protocol is highly impacted by this approach, and all necessary measures were followed to preserve privacy and the individuals' identities.

In a distributed environment, each party is considered either semi-honest or malicious. Semi-honest participants follow the protocol specifications, but are curious to learn more information. However, they do not deviate from the execution of the protocol. Conversely, malicious participants are categorized into internal and external. Internal adversaries deviate from the protocol, for example, by sending specific inputs, with the main purpose of discovering other parties' private data. External adversaries will try to impersonate a legal participant and then behave as an internal adversary. In the current protocol, both adversary types are considered.

All participants, the miner and each party, undertake the process of authentication, so they can mutually recognize if they are connected to a secure and literal participant. Each participant sends their digital signatures, assuming they were

Algorithm 1 Protocol for nominal attribute values

for $c_1 \dots c_m$ class value **do**
  for $a_1 \dots a_i$ attribute value **do**
    for $1 \dots n$ party **do**
      1. compute # instances f_{im} with attribute value i and class value m
      2. compute # instances f_m^n with class value m
    end for
    Miner applies the homomorphic primitive:

$$E(f_{mi}^1 \otimes f_{mi}^2 \otimes \cdots \otimes f_{mi}^n) = E(f_{mi}^1) \otimes E(f_{mi}^2) \otimes \cdots \otimes E(f_{mi}^n)$$

$$E(c_m^1 \otimes c_m^2 \otimes \cdots \otimes c_m^n) = E(c_m^1) \otimes E(c_m^2) \otimes \cdots \otimes E(c_m^n)$$

  end for
  Miner computes:

$$P_{im} = \frac{E(f_{mi}^1 \otimes f_{mi}^2 \otimes \cdots \otimes f_{mi}^n)}{E(c_m^1 \otimes c_m^2 \otimes \cdots \otimes c_m^n)}$$

end for

Algorithm 2 Protocol for numeric attribute values

1: **for** $c_1 \dots c_m$ class value **do**
2: **for** $1 \dots n$ party **do**
3: **1.** compute # instances f_m with class value c_m
4: **2.** compute sum of instances s_m^n with c_m
5: **end for**
6: Miner computes using homomorphic primitive:
7: Total sum s_m :

$$E(s_m^1 \otimes s_m^2 \otimes \cdots \otimes s_m^n) = E(s_m^1) \otimes E(s_m^2) \otimes \cdots \otimes E(s_m^n)$$

8: Total # instances N_m :

$$E(f_m^1 \otimes f_m^2 \otimes \cdots \otimes f_m^n) = E(f_m^1) \otimes E(f_m^2) \otimes \cdots \otimes E(f_m^n)$$

9: Mean:

$$\mu_m = \frac{s_m}{N_m}$$

10: **end for**
11: **for** $c_1 \dots c_m$ class value **do**
12: **for** $1 \dots n$ party **do**
13: **for** instance y **do**
14:

$$u_{mn}^i = x_{mn}^i - \mu_m$$

15:

$$u_{mn}^i = \sum_y (u_{mn}^2)$$

16: **end for**
17: **end for**
18: Miner compute variance:
19:

$$u_m = E(u_m^1 \otimes u_m^2 \otimes \cdots \otimes u_m^n) = E(u_m^1) \otimes E(u_m^2) \otimes \cdots \otimes E(u_m^n)$$

20:

$$\sigma_m^2 = u_m * \frac{1}{N_m - 1}$$

21: **end for**

signed by a certification authority (CA), to confront such behaviors. This operation ensures that only authorized parties participate in the protocol and they are assured that a connection with the actual miner was accomplished.

Privacy is preserved only if confidentiality, anonymity, and unlinkability are fulfilled. All transmitted messages between each party and the miner are encrypted, and a message is only decrypted by the party that was supposed to receive the message. The homomorphic primitive ensures that the miner is unable to identify the inputs each party forwards, accomplishing anonymity and unlinkability. Both the identity and the private data of each party remain secret. In the proposed protocol, integrity mechanisms are exploited to identify any modification carried out by active attackers, with the prime goal of diminishing the accuracy of the final outcomes or discovering sensitive data. An SHA-1 digest is concatenated to every transmitted message, prohibiting these behaviors and assuring any altered message will be detected. Section 4.4 describes in depth the security and threat model of the proposed protocol.

The proposed protocol satisfies the following main requirements, to ensure better performance in scalable and distributed databases:

- Data mining processes extract statistical information.
- Database records are horizontally or vertically partitioned.
- Data can be either nominal or numeric.
- A large number of parties can be handled.
- Only authorized parties can send inputs to the miner.
- The communication among parties is not feasible.
- The miner must be connected with at least three parties before proceeding to the mining process.
- The miner collects all the messages encrypted and performs the mining process.
- Individual records remain secret and only overall results are revealed.
- Any data given as input includes the encrypted frequency of each attribute value in relation to any class value and cannot be modified, reduced or copied.
- A summary is concatenated to each transmitted message, as a result of applying the one-way hash function SHA-1.
- It is essential that computation and communication costs are low, both for each party and the miner.

4.2.2 Protocol analysis

The protocol presented in the current work follows the classical homomorphic election model, in particular, an extension for supporting the multi-candidate election scheme, where each party has k-out-of-1 selections [50]. The Paillier cryptosystem follows the homomorphic model and preserves privacy while mining operations

are applied in a fully distributed environment. A data collector—in the current proposal, the miner—collects and organizes all data forwarded by the participants of the protocol. The miner exploits the homomorphic primitive when all encrypted data are collected, and applies the tree augmented naive Bayesian classification model. Through the classifier, correlations among the attributes are generated, resulting in the creation of a network structure that represents them. Each transmitted message during the execution of the protocol includes an SHA-1 digest to confirm that any modification has not been performed. The miner delivers the final results to each party who contributed to the creation of the mining model. The frequency of each attribute value in relation to each class value, for both horizontally and vertically partitioned databases, constitutes the final results, and we assume that every party is aware of the class value for vertically partitioned database records.

The protocol is divided into six main phases and applied for both horizontally and vertically partitioned databases. The protocol notations are given in Table 4.1.

Table 4.1: Protocol notations

S_{pu}	Miner's public key for encryption/decryption
S_{pr}	Miner's private key for encryption/decryption
C_{pu}	Party's public key for encryption/decryption
C_{pr}	Party's private key for encryption/decryption
S_{Dpr}	Miner's private key for digital signature
S_{Dpu}	Miner's public key for digital signature
C_{Dpr}	Party's private key for digital signature
C_{Dpu}	Party's public key for digital signature
$H(m)$	SHA-1 hash of message m
$Enc(m)k$	Encryption of message m with key k
$Decr(m)k$	Decryption of message m with key k
A_i	Database Attribute
M	Random variable

Phase 1. Key generation The miner generates the encryption key pair (S_{pu} and S_{pr}) through the Paillier's cryptosystem key generation phase. The miner produces a 1024 bit digital signature key pair (S_{Dpu} and S_{Dpr}) with the Rivest–Shamir–Adleman (RSA) cryptosystem using the MD5 hash function. We assume that each party is able to obtain the public keys. The same procedures are followed by each party who also create the encryption key pair C_{pu}/C_{pr} and an RSA key pair C_{Dpu}/C_{Dpr}) (Figure 4.3). We again assume that the miner is as well able to obtain the public keys of all parties. In the key establishment phase, the miner also generates and forwards a random value M.

Phase 2. Mutual authentication The miner and each party that participates in the protocol are mutually authenticated by exploiting the digital signature scheme (RSA), as each participant possesses a private and public key pair. This key pair is generated and used only in this phase of the protocol assuming it was signed by a CA. We assume that all parties are able to obtain the public keys of the other participants.

If a party requests to connect with the miner, during the authentication phase, they forward the public key C_{pu} and the digital signature, encrypting the C_{pu} key with the miner's C_{Dpr} private key. The miner proceeds to the decryption of the digital signature with the public key C_{Dpu} of the party and generates a digest of the C_{pu} message. If the miner is able to verify that the party is able to participate in the protocol, responds by sending his public key S_{pu} and digital signature encrypted with the S_{Dpr} private key. The party continues with the same procedure by decrypting the miner's digital signature with public key S_{Dpu} and creates a digest of the S_{pu} message (Figure 4.3). After these steps are completed, the party is assured that a verified connection with the actual miner is achieved, and both the miner and each party have access to and participate in the protocol, excluding any unauthorized participants. After exchanging all keys, every transmitted message is encrypted. The next step is to send the random variable M, which is used by the Paillier cryptosystem, to confront any chosen-plaintext attacks. This variable is sent encrypted with each party's public key C_{pu}.

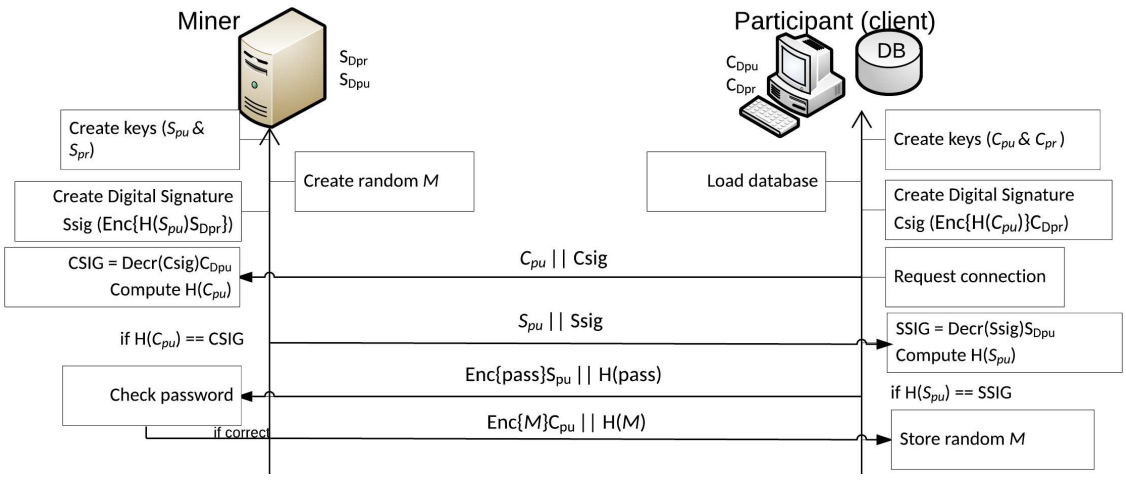


Figure 4.3: Key generation and mutual authentication phases.

Phase 3. Data collection After all the above procedures are performed, the miner is ready to accept the participant's personal data. A party can participate in the exportation of statistics, providing their own sensitive data. However, the data contained in the database cannot be disclosed in the notion of verbatim records, neither to the miner nor to other participants nor to any attacker not involved in the protocol. Every record is examined for the presence of missing values.

The collection of data begins from the miner. If a party consents to the creation of the classification model, they initially send every possible value of the class and every possible attribute value. All messages sent are encrypted with the miner's public key S_{pu}. For horizontally partitioned databases, each party sends all possible attribute values. For vertically partitioned databases, each party sends only the values of the attributes that possess the required attribute; if the party does not possess the requested attribute, A_i returns zero. The miner is not aware of the possession of individual values at the end of this step.

The miner requests the frequencies for attribute A_i for each connected party (Figure 4.4). Using the miner's public key S_{pu}, each party forwards the frequency of each value for A_i attribute in relation to every class value, encrypted. The only sensitive data sent by all parties are these frequencies, and they are encrypted. Because the homomorphic primitive is applied, the miner remains unaware of the specific frequencies. These procedures are necessary for the miner to initialize the classifier.

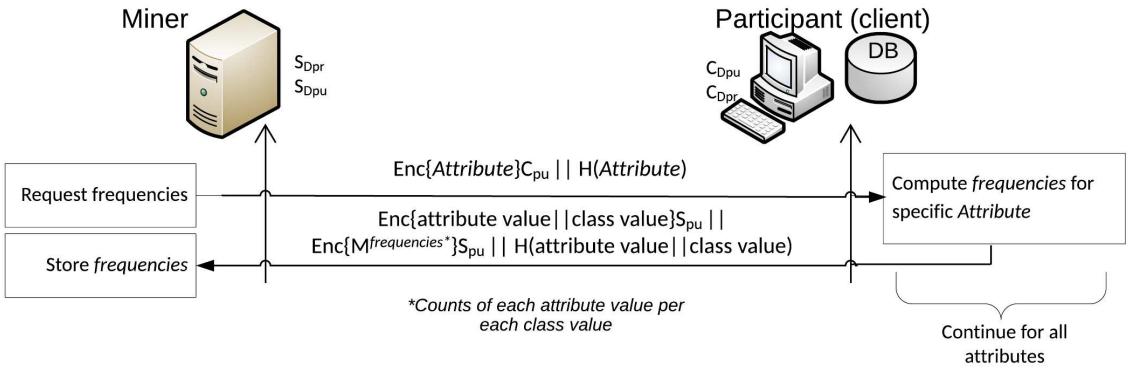


Figure 4.4: Data collection phase.

Phase 4. Classifier Initialization If the Miner has collected the encrypted frequencies related to attribute A_i from all three parties, applies the homomorphic primitive. All encrypted frequencies are decrypted simultaneously, and the miner obtains the overall distributions of each A_i attribute value in relation to each class C value. The process continues with the miner requesting the frequencies for the next A_{i+1} attribute. The process is completed after the collection of all frequencies for all attributes A_n. For horizontally partitioned databases, n represents the total number of attributes. For vertically partitioned databases, n refers to the sum of each party's number of attributes. The classifier initialization is successful when at least three parties cooperate in the implementation of the protocol (Figure 4.5).

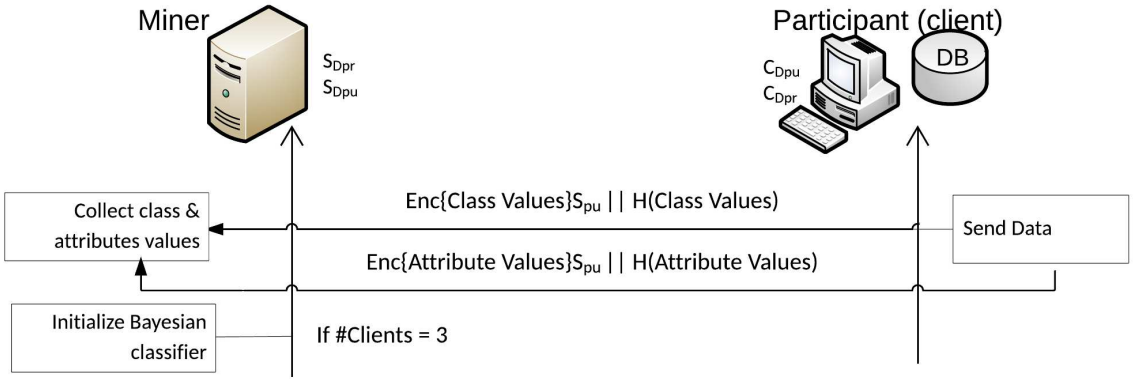


Figure 4.5: Classifier initialization phase.

Phase 5. TAN classifier creation The miner can proceed to the creation of the TAN classifier after the classifier initialization phase is complete, meaning all frequencies are collected and decrypted for each attribute, from at least three participants. As described in Section 4.1.3, the miner now is in the position to create the tree augmented naive Bayes model (Figure 4.6).

Phase 6. Final results When all the above-mentioned phases are complete, the final results of the mining process are delivered by the miner. The miner sends the results to each party involved in the creation of the data mining model, encrypted with their own public key C_{pu} (Figure 4.6).

After the creation of the mining model and the shipment of the final results, every participant can request that the miner respond with the class value and the corresponding possibility that accrues from a set of possible attribute values, classifying new instances. This process was used to evaluate the performance of the classifier, and the results are presented in the next section. Figure C.1 in Appendix C presents in sequence the proposed protocol. The client interface is presented in Figure C.2 in the same appendix.

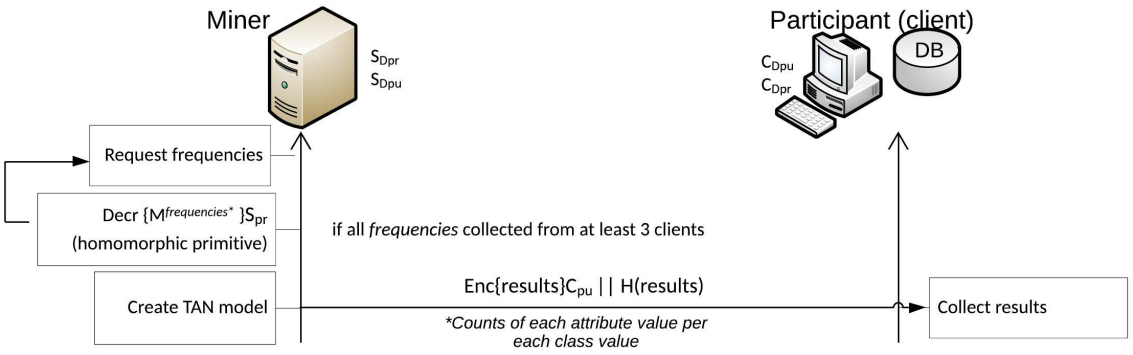


Figure 4.6: TAN classifier creation and final results phases.

4.3 Protocol evaluation

In this section, we evaluate the proposed protocol in terms of security and computational cost. Primarily, the mean Paillier key pair generation time was estimated for both the miner and the party, and compared with El-Gamal key generation. We compared the mean time needed to create the digital signatures in two different systems. The main procedures of the protocol were examined to demonstrate that they have a fast computation time while preserving privacy. Three different scenarios were established for this purpose. The cryptosystem performance was evaluated on encryption and decryption run times. The TAN classifier was evaluated using recall, precision and F1 score variables as metrics.

All experimental results were calculated and are presented in milliseconds (ms). Most experiments were conducted on a modest PC with Intel i5 2.4 GHz with 4 GB of RAM. To extend some experiments, we performed them using a more advanced computer. The purpose of the second system was to evaluate if a more advanced system can decrease the computational cost of specific phases of the proposed protocol, like the key generation phase. The new PC was equipped with Intel Core i7, 2.9 GHz, and 16 GB of RAM, and each phase in which this computer was used is denoted as i7 in contrast to the computer with the i5 processor. For this study, only the key establishment experiments were conducted in both systems. The proposed protocol was implemented in Java programming language, and both the miner and all three participant interfaces were running on the same system.

The experiments showed that the performance of the protocol is mainly shaped by the data collection phase, which is proportional to the number of attributes included in the databases. We conclude that the partition of databases affects the collection of data phase mainly when the amount of instances increases.

4.3.1 Key establishment

The key establishment was evaluated on both systems described above. Measurements were collected from 50 runs performed for one participant and the miner to calculate the performance of the key generation, authentication, and login operations. The encryption key pair generation and the RSA digital signature creation were included in the key generation phase. We assumed that each participant knew the miner's S_{Dpu} key and the miner was aware of all public keys C_{Dpu} of the parties involved in the mining process.

From the experiments conducted on the i5 computer system, we found that a party requires 479 ms to create the encryption key pair and 122 ms to generate the digital signature. The miner performs the encryption key pair generation in 433 ms and requires 108 ms to create the digital signature. The random variable M used by the Paillier cryptosystem was produced in 43 ms. When the experiments were conducted on the i7 computer system, the mean times significantly improved, as shown in Figure 4.7. The Paillier encryption key generation mean time was almost four times faster when the measurements were performed in the i7 system. The El-Gamal key generation was implemented to compare this phase using the two cryptosystems. The Paillier and El-Gamal key generation experiments were performed on the i7 system and we found that the key generation of the El-Gamal cryptosystem was remarkably slower compared to the Paillier cryptosystem. The

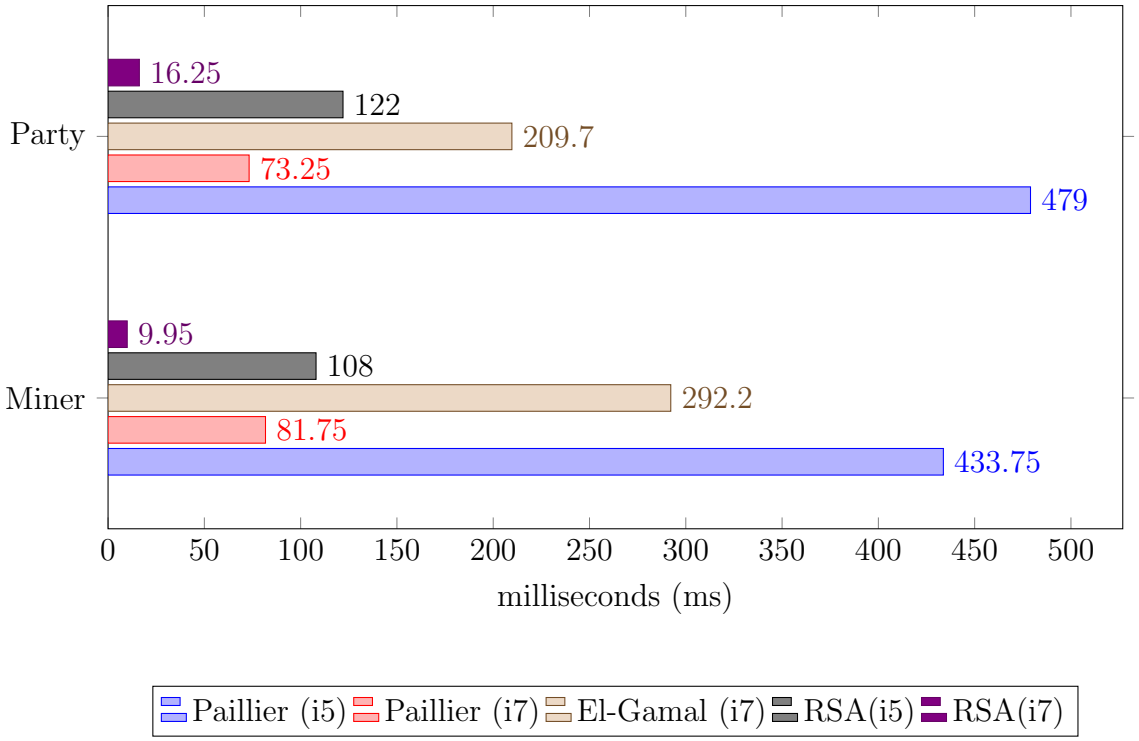


Figure 4.7: Comparison of key establishment procedures.

generation of RSA digital signatures was also compared between the two computer systems, as presented in Figure 4.7. The digital signatures generation was significantly faster when the computer system was more advanced (i7). As shown by the results, the Paillier asymmetric encryption algorithm is efficient in terms of key establishment.

The time needed by the miner and each party to be mutually authenticated is represented by the authentication time. In this phase, each participant sends the public keys and digital signatures created in the key establishment phase. The measurements from the conducted experiments showed that the mutual authentication is achieved in 24 ms. In the login phase, the party sends the miner's password encrypted with the S_{pu} key and the miner, in return, responds with the correctness of the password received by sending the encrypted random variable M. The mean login time (262 ms) is longer than the mean authentication time as all the messages transmitted are encrypted, meaning decryption and encryption operations are required.

4.3.2 Experiments

The performance of the proposed protocol was measured by separately examining its main procedures:

1. the collection of data from the miner (DC),
2. the initialization of the classifier (CI),
3. the creation of TAN classifier (TAN CC) and
4. the delivery of the final results (FR) to each party.

Table 4.2 presents the three customized scenarios used for conducting the experiments based on the database partition. For each scenario, three parties were connected to the miner and participated in the protocol with either horizontally or vertically partitioned databases. These scenarios were evaluated and compared to determine the performance of the protocol when different amounts of records and attributes are involved in the creation of the mining model.

Table 4.2: Experiment scenarios

(a) Horizontally partitioned

	Records	**Attributes**
Scenario 1	50	5
Scenario 1	100	5
Scenario 3	100	10

(b) Vertically partitioned

	Records	**Attributes**
Scenario 1	50	3
Scenario 2	100	3
Scenario 3	100	6

The experiments were performed using real datasets provided by the UC Irvine Machine Learning Repository [89]. The data were tailored for each scenario, and the training set size was set to 1000, 2000, and 5000 records. A simplified structure of this dataset is displayed in Figure 4.8. Table 4.3 provides the mean time to complete each phase of the proposed protocol.

The customized scenarios were selected to compare the performance of the protocol depending on the number of attributes and records. From the results, we found

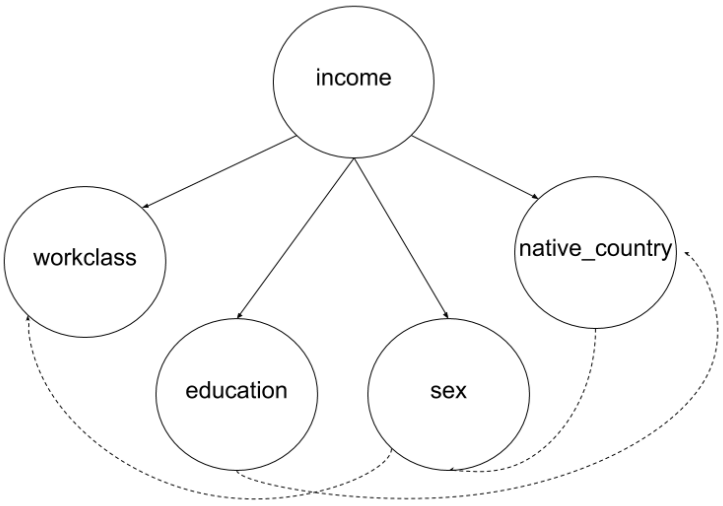


Figure 4.8: Simplified TAN structure of "Adult" dataset.

that the overall time to complete the main procedures of the protocol is mainly determined by the data collection phase, which mostly increases with increasing number of attributes. Comparing both database cases, we found that the partition affects mainly the data collection phase by doubling the mean time, mostly as the number of instances is increased.

The distributed environment with three parties connected to the miner was selected because we wanted the first evaluation of the protocol to be less ambiguous. If more than three parties are connected with the miner and send their data, the data collection phase is expected to be affected as well. In the future, conducting experiments with more parties involved can prove the scalability and efficiency of the proposed protocol.

4.3.2.1 Experiments: Horizontally partitioned databases

The scenarios used to evaluate the protocol for horizontally partitioned databases are presented in Table 4.2a. For the first scenario, each database consisted of 50 records and 5 attributes; in the second scenario, it consisted of 100 records and 5 attributes; and in the third scenario, 100 records and 10 attributes. The results showed that the initialization of the classifier has a low mean time, but it is affected when the number of attributes is increased. Conversely, the initialization time is slightly longer as the amount of database instances increases. Similar conclusions were drawn during the data collection phase. However, the data collection process has a long execution time, as each party has to send all their data/frequencies to the miner. The data collection time increases reasonably when the number of instances is higher, but when the database consists of a larger number of attributes, the miner requires more time to collect all the frequencies. The mean time to create the TAN model increases when the quantity of instances increases, unlike the increase in the mean time when the attributes are doubled. Increases in the number of attributes do not influence the mean time. When both the quantity of instances and attributes increase, the mean time to forward the final results to each party also increases.

Table 4.3: Main procedures comparison for each scenario

(a) 1st scenario results

Procedure	**1st horizontal**	**1st vertical**
DC$^{\bullet}$	31777	58939
CI*	13	57
TAN CC$^{\diamond}$	39	52
FR†	2407	3411

(b) 2nd scenario results

Procedure	**2nd horizontal**	**2nd vertical**
DC$^{\bullet}$	35502	59764
CI*	16	56
TAN CC$^{\diamond}$	17	118
FR†	3744	3592

(c) 3rd scenario results

Procedure	**3rd horizontal**	**3rd vertical**
DC$^{\bullet}$	94793	89073
CI*	30	64
TAN CC$^{\diamond}$	68	110
FR†	4455	6076

$^{\bullet}$ *Data collection.*

* *Classifier initialization.*

$^{\diamond}$ *TAN classifier creation.*

† *Final results.*

4.3.2.2 Experiments: Vertically partitioned databases

The scenarios used to evaluate the protocol for vertically partitioned databases are presented in Table 4.2b. In each customized scenario, we assumed that all parties involved know the class C value. For the first scenario, each database included 50 records and 3 different attributes (plus the class attribute); in the second scenario, the number of records was doubled and the number of attributes remained the same; while in the third scenario, 100 records and 5 different attributes were included in each database. The results showed that all the procedures of the protocol require slightly more time to be completed compared to the corresponding scenario for horizontally partitioned databases. The collection of data requires almost twice the time due to the data partition. Like with horizontally partitioned databases, the creation of the TAN classifier and the data collection phases require more time when the amount of instances increases. When the attributes increase, the data collection time lengthens, but less time is required in comparison to the horizontally partitioned databases. The results showed that the classifier requires more time to be initialized in relation to horizontally partitioned databases. If the number of attributes increases, the TAN classifier creation behaves similarly for both horizontally and vertically partitioned databases. The delivery of the final results is slower for double the number of attributes for vertically partitioned databases.

4.3.3 Cryptosystem performance

The mean encryption and decryption time were calculated to measure the performance of the Paillier cryptosystem. During the execution of the protocol, different messages were transmitted, each one with a different number of characters. From all the above executed scenarios, we measured all the encryption and decryption mean times.

The results showed that a message can be encrypted in 51.5 ms on average. The average time needed to decrypt a message was similar, and the measurements revealed that a message can be decrypted in 67 ms. The decryption time slightly increased most probably because of the application of the homomorphic primitive. These results are collected from the 'i5' system. By applying the same experiments in the 'i7' system, the encryption time is 11.6 ms and 19.4 ms is the decryption time. As expected in the newer system the measurements are lower, almost 1/4 reduced. We conclude that the Paillier cryptosystem is efficient as the mean times are low. In the future, a comparison of the Paillier and El-Gamal cryptosystem would determine the most appropriate algorithm in terms of computational cost.

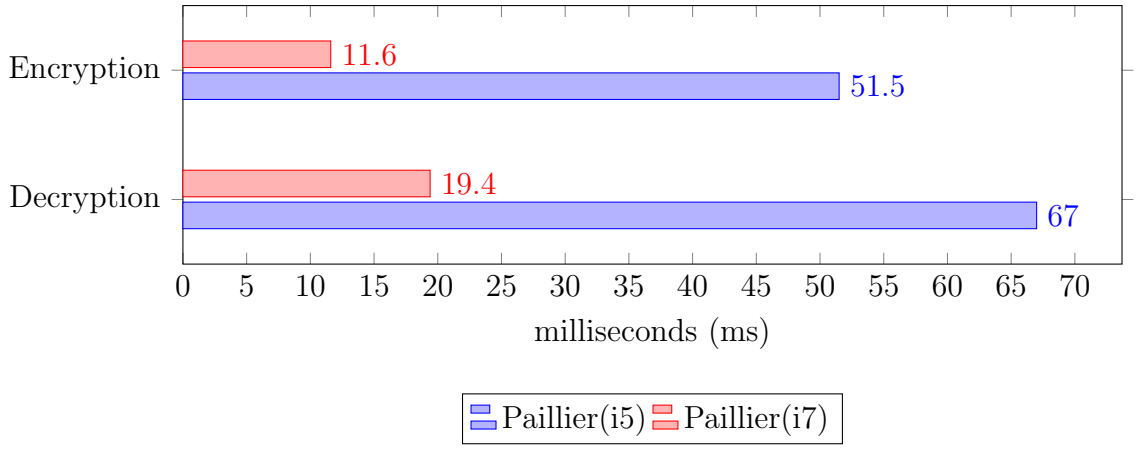


Figure 4.9: Cryptosystem performance

4.3.4 Classifier evaluation

To examine the mining model created by the miner, we calculated the recall, precision, and F1 score. The percentage of records categorized with the correct class in relation to the number of all records with this class is the recall. The percentage of records that truly have a certain class over all the records that were categorized with this class is the precision. The F1 score is computed using Equation (4.10). If the F1 score is equal to 1, the precision and recall results are perfect. The lowest possible F1 score is 0 if either the precision or recall is 0.

$$F_1 = 2 * \frac{precision * recall}{precision + recall} \tag{4.10}$$

Three customized datasets with different amounts of instances were used as training sets (1000, 2000, and 5000 records). The databases were obtained from a real dataset [89] and contained 14 attributes. A test set of 100 records (10% of the training records) was used, which was not included in the training phase. The aim of the classifier evaluation is to determine which mining model correctly classified the test set. The evaluation results of the TAN classifier are presented in Table 4.4. The naive Bayes classifier evaluation results are presented in Table 4.5.

Comparing the two classifiers, we found that TAN correctly classified more instances compared to the naive Bayes classifier. Analyzing all three measurements mentioned above, we found that the TAN classifier is a more accurate and appropriate method compared to traditional naive Bayes.

Table 4.4: TAN classifier evaluation results.

Records	1000		2000		5000	
Correct	54		55		56	
Incorrect	46		45		44	
Class value	≤ 50	> 50	≤ 50	> 50	≤ 50	> 50
Recall	0.42	0.63	0.52	0.6	0.54	0.6
Precision	0.48	0.57	0.73	0.38	0.73	0.39
$\mathbf{F_1}$	0.448	0.5985	0.6074	0.4653	0.6208	0.4727

Table 4.5: Naive Bayes classifier evaluation results.

Records	1000		2000		5000	
Correct	49		49		50	
Incorrect	51		51		50	
Class value	≤ 50	> 50	≤ 50	> 50	≤ 50	> 50
Recall	0.42	0.54	0.48	0.52	0.50	0.8
Precision	0.43	0.53	0.77	0.23	0.47	0.2
$\mathbf{F_1}$	0.42495	0.5350	0.59136	0.3189	0.4845	0.32

4.4 Threat model

Many serious attacks need to be considered when a protocol is being developed. Distributed environments have to prevent every possible threat on systems designed with privacy preservation as their main concern. A threat is a potential violation of security that exists when an action could breach security and cause harm. A threat can be either intentional (an individual attacker) or accidental (a computer malfunction).

Some types of security threats are related to unauthorized access. Services or data becoming unavailable can be considered another security threat. The modification of transmitted data is considered a major threat to the security of a system, as well as the generation of fabricated data [90]. This section presents and discusses the possible threats that can be confronted by the proposed protocol. Table 4.6 summarizes these attacks and how they are approached and solved using appropriate mechanisms by the presented system.

Security in distributed environments is an important concern that needs to be analyzed to discover possible vulnerabilities or threats and avoid information loss. A distributed system must follow some requirements for security enforcement [90]:

Table 4.6: Possible security threats and their confrontation.

Attacks	**Security mechanism**
Eavesdropping	Asymmetric Cryptography (Paillier)
Collusion	Lack of communication among parties
Probing	Three parties, Cannot send blank input
Man-in-the-middle	Digital Signatures
Message modification	SHA-1
Denial of Service (DoS)	Data send once
Chosen Plaintext	Random variable M

- The sender of a message should be able to know that the message was received by the intended receiver;
- The receiver of a message should be able to know that the message was sent by the original sender;
- Both sides should be guaranteed that the contents of the message were not modified while transferring data.

There are some broad areas of security in distributed systems: authentication, access control, data confidentiality, data integrity, encryption, digital signature, and nonrepudiation. Authentication is a fundamental concern when developing distributed systems. All entities in a secure system should follow an authentication process assuring the communication is authentic. The authentication service assures the participants that the message received is actually from the stated source. This process occurs the first time a connection is initialized, and assures that all entities involved are authentic. It must also ensure that there is no interference by unauthorized third parties. Access control is the ability to control the access to systems and prevent the unauthorized use of a service. This is achieved by identifying or authenticating each participant that tries to gain access, so that specific access rights are provided to each party. Confidentiality is the protection of data being transmitted from attackers and unauthorized disclosure. There are several levels of protection, both regarding the data content being sent and the data flow. This requires the attacker to not be able to observe the source and destination or other characteristics of the traffic flow. As with confidentiality, data integrity mechanisms can be applied to part of a message or the whole message. The most useful approach is full-message protection, ensuring messages received are not modified. Encryption mechanisms transform data into a form that is not readable without the use of intelligent systems. The transformation and recovery of data depend on the combination of algorithms and encryption keys. Digital signatures allow the recipient of a message to prove the source and integrity of the message and protect against forgery. The digital signature can be signed to produce digital certificates

that establish trust among users and organizations. Nonrepudiation prevents users from denying they received or send a transmitted message. In these cases, the messages are registered by a notary so that none of the participants can back out of a transaction and disputes can be resolved by presenting relevant signatures or encrypted text [91]. In the present work, we did not consider nonrepudiation, as these cases fall beyond the scope of the presented protocol.

All parties involved in a distributed environment are considered to be mutually mistrustful and, in some cases, curious to learn information about other participants' data. Every participant is considered either semi-honest or malicious:

1. Semi-honest adversaries follow the protocol specifications; they do not collude but are curious to discover other party's data during the execution of the protocol.

2. Malicious adversaries can be internal or external. Internals deviate from the protocol and send specific inputs to infer other participants' private data. An external adversary tries to impersonate a legal participant and behave as an internal.

The miner could be considered an internal adversary. To address such behaviors in our proposal, external adversaries were excluded as they cannot participate, since all parties have to send their digital signatures. We assumed the digital signatures are signed by a certification authority. The mutual authentication provided by the proposed protocol excludes any unauthorized users. Participants with no permission to connect with the miner are not able to participate in the protocol. This means that the Miner cannot be an internal adversary, as all participants are aware if a connection is established with the actual miner. Participants who also behave as internal adversaries are restricted to sending blank inputs or missing values to the system. The only information revealed are the final outcomes; further information is impossible to obtain. By exploiting the digital signatures, man-in-the-middle attacks are not possible.

Several studies have examined the re-identification attack on privacy-preserving data mining algorithms. Many hospitals, for example, are willing to publish their data for research on the condition that any identifier that allows information pertaining to specific patients is removed, either for administrative or commercial reasons. This action, however, may not be enough, as re-identification attacks can lead to different public databases, thus revealing the real names of the referring patients [31]. To reduce re-identification risk, in the proposed method, we consider the privacy of the individuals: the data are anonymized and the final results are published to each participant to prevent any possibility of private and identification data being revealed.

Some security attacks depend on the presence of one or more miners in a distributed environment or personal data being transmitted among two or many parties. In distributed environments with only one miner, the final results can be discovered by the data collector, but if more miners are involved, the protocol is vulnerable to collusion attacks. If parties directly exchange data with each other, in the two-party model, each party can easily determine the other party's private data. In a model where multiple parties are connected without a miner as the data collector, malicious parties can modify the input data, which can be disastrous if $n - 1$ users collaborate. In the proposed protocol, to prevent these behaviors, data are exchanged

only between the miner and each party, ensuring there is no collusion among the participants. At least three parties must be involved, preventing any probing attack. This approach establishes a secure protocol and semi-honest adversaries are faced as the only information revealed and sent by the miner is the final outcomes. We did not consider the collaboration of the parties outside of the protocol. Participants in a protocol have a mutual interest to follow the protocol's principles in real-world applications.

If the requirements of confidentiality, anonymity, and unlinkability are fulfilled, privacy can be preserved. The Paillier cryptosystem ensures that sensitive data remain secret. The asymmetric encryption establishes an environment in which all parties receive the messages that were intended only for them, and they are the only ones that can decrypt these messages. Eavesdropping attacks or data leaking are successfully managed by the proposed protocol. In addition, the Paillier cryptosystem exploits the homomorphic primitive for both nominal and numeric attribute values, which guarantees that the original data will not be revealed to any attacker, the participants, or the miner. This primitive achieves anonymity and unlinkability, two aspects that the proposed system is committed to providing. The Paillier cryptosystem is vulnerable to chosen-plaintext attacks. This type of attack is overcome by the current protocol using a random variable (M).

If active attackers try to modify any message exchanged during the execution of the protocol and alter the final results or disclose sensitive data, they are stopped using integrity mechanisms (SHA-1). The participants in the proposed protocol are unable to resend their data and the protocol can be executed only once per computer system, preventing denial of service attacks. Blank or missing inputs are also excluded. Table 4.7 summarizes the security requirements and the technique used in the current protocol.

Table 4.7: Security requirements.

Requirement	**Technique**
Mutual authentication	Digital Signatures, Password
Confidentiality	Paillier cryptosystem
Anonymity	Homomorphic primitive
Un-linkability	Homomorphic primitive
Integrity	SHA-1 hash function

Chapter 5

Summary

Voluminous data stored in distributed databases are exchanged daily due to technological progress. Global information can be acquired and important patterns can be detected by applying data mining techniques on statistical databases. Such databases often contain private data, and their disclosure when mining operations are applied could compromise the privacy and the fundamental rights of individuals.

The proposed protocol in Part I focuses on solving this problem. A properly designed privacy preserving data mining technique was developed for a distributed environment. Participating databases can be horizontally or vertically partitioned, supporting both nominal and numeric attribute values. A data collector, the miner, groups the data received by at least three parties and performs all the operations to generate the mining model. Communication among parties is infeasible and the only workflow is between the trusted data collector (miner) and each participant in the protocol. All messages exchanged during the execution of the proposed protocol are encrypted using the Paillier cryptosystem. The homomorphic primitive ensures that the miner decrypts the messages received all at once, preserving the privacy of data. Cryptography-based techniques, as shown by previous research, are the most appropriate approaches in terms of accuracy, as the original data are not modified or transformed; therefore, the quality of the final results remains high. All transmitted messages are examined for any type of modification, as each message is concatenated with its summary produced by the one-way hash function SHA-1.

The experimental results showed that the proposed protocol is effective and efficient for both database partitions. The performance of the protocol is mainly affected by the increase in database attributes. Yet, given the size of the real dataset, this is considered acceptable.

Most of the privacy preservation methods designed for data mining purposes are theoretical and not implemented to support real world applications. The contribution of the proposed protocol is significant as, to the best of our knowledge, none of the previously proposed techniques was designed and implemented for both horizontally and vertically partitioned databases while simultaneously providing accurate results and preserving privacy.

Part II

Privacy preservation of social networks

Chapter 6

Introduction

Social networks are exploited and analyzed by many different research fields such as sociology and psychology. Their increasing popularity has raised the interest of researchers also in the data mining community. Since social networks are being released to the public for research purposes, there is an increasing concern about the privacy of the individuals involved [92, 93]. Therefore, it is necessary before the network data being published for analysis, data mining or other purposes, to ensure that these data do not contain any sensitive information such as the identities of the individuals involved and their relationships [94]. Privacy risks have been studied for anonymized social networks, and the results have shown that social networks need to be anonymized such that sensitive data as protected but at the same time the utility is preserved [95]. The action of removing only the identifiers or by replacing them with other unique identifiers before releasing the network data to the public is not sufficient [3].

A social network and every network can be modeled as a graph, where nodes denote entities from the real world such as individuals and organizations. Individuals in a social network might have stronger or weaker social ties with other individuals in the network [96]. These strong or weak relationships between individuals are represented by edges. A graph model may include additional information about the involved individuals and their relationships. These additional information can be for example the quantification of the strength of each connection which is denoted as edge weights. Moreover, further information can be the node attributes which express user information such as preference and affiliation. For example, the edges and edge weights in [97] indicate the social interaction and the strength of the relationships among the members of a Karate club at a US university.

A network includes usually three different entities: the users of the network whose identity and private data need to be protected, and they are represented by nodes in a graph; the adversary that wants to acquire sensitive information by possessing background knowledge or by combining the released graph with external information; the analyst whose adjective is to extract useful information by analyzing the released graph.

Due to social networks being released, there is a growing concern about personal privacy being breached. Thereby effective and efficient anonymization techniques that allow analysis of the graphs are required. A privacy breach occurs when sensitive information about the user is disclosed to an adversary. The privacy model involves three components: a specification of what is considered private and needs

to be anonymized, the external information that an adversary may possess and a set of measurements on how much private are the data released and the loss in their utility for analysis.

Nevertheless, the existence of edge weights pose additional challenges in anonymizing a network. An adversary may possess supplementary information about the edge weights of the individuals, but any anonymization method should still be able to protect the individuals. Methodologies that have been designed specifically for unweighted graphs are not sufficient enough to protect weighted graphs from attackers. In case an adversary has additional information related to the edge weights, there is a need to actually prevent from their disclosure [98]. Edge weights are important for many analyses of weighted graphs, but at the same time as much information as possible should be maintained, while preserving the utility of the anonymized data [95].

If we consider the social network to be a weighted graph, then the privacy breaches in social networks can be categorized into three types: identity disclosure, link disclosure, and content disclosure [99–101].

- Identity disclosure is a fundamental privacy issue in social networks, in which specific individual identity is revealed because an adversary was able to associate a node of the graph to this individual. The identity disclosure is considered as the key of privacy violation in social networks because it usually leads to the disclosure of content information as well as information about their relationships.

- Link disclosure occurs when the existence of a relationship between two individuals is discovered. An adversary may want to know the degree of relationship between two entities. If the relationship of two individuals can be determined by a certain path, then the privacy is compromised.

- Content disclosure is a privacy breach that occurs when data associated with a node or an edge are revealed. These data can be either associated with an individual or its relationships. Data such as the edge weight can be as well considered sensitive in a weighted graph.

Additional privacy breaches can also be the disclosure of node existence, determining if a target node appears or not in the network, the disclosure of node and edge attributes which are considered private, the properties disclosure regarding the network structure around an individual, such as the degrees clustering coefficient or properties of the neighbors of a node [102].

An anonymized network can be evaluated with regards to two criteria on whether (i) the private data of each user are protected and (ii) the utility of the graph is preserved. Since published data are used for analysis, anonymization must ensure a balance between the privacy of the individuals, their connections and their related content, and the utility of the resulting data such that it is not compromised. For example, a social network represents the residents of a small city in which some people got hospitalized with covid-19 virus. The nodes represent the people who got affected by the virus, the links show the relationships with their immediate environment, and the edge weights stand for the frequency of communication between them. This network must be anonymized so that it is not possible to later identify

the patient's identity and the strength of their relationships. Such a network can be analyzed by researchers to identify how the frequency of their contact affects the health of the patients, therefore the anonymization technique must ensure the statistical properties are preserved.

6.1 Anonymization techniques

Anonymization techniques that are proposed for privacy preserving data mining of graphs and networks have to face the following challenges: how to model the privacy information that might be under attack; how to model the background knowledge that an adversary may possess to attack the privacy of a target; how to model data utility and information loss and how to develop an efficient anonymization method that will preserve data utility, by minimizing information loss [103]. Several techniques that have been proposed to preserve privacy in relational data are not applicable to social network data, and cannot be used for protecting privacy straightforwardly. The anonymization of social networks is far more challenging compared to anonymizing tabular data [104].

Data should be anonymized properly before releasing in order to preserve privacy. The anonymization techniques should consider both the privacy and utility of the data. For example, naive anonymization removes all identifiers from the original graph and replaces them with random numbers in the released graph. However this method may be insufficient. By applying this method the utility of the data is highly preserved, but the released network is vulnerable to attackers who by being aware of the network structure can be able to re-identify an individual [105]. An adversary can compromise privacy by combining external information with the released graph, de-anonymize the nodes and learn the existence of the relationships between the de-anonymized individuals [93].

Clustering-based and modification methods are the two state-of-the-art categories of anonymizing social network data [102].

Clustering-based or generalization approaches cluster nodes and edges into groups and replace a subgraph with supernodes and superedges. With this approach, all information related to an individual is properly hidden. This category can be divided into node clustering [105], edge clustering [100], node and edge clustering [106] and node attribute mapping clustering methods [107].

Modification approaches modify the graph by inserting or deleting nodes and edges in a graph, either by directly adding or removing specific edges or by randomly adding or removing edges. Modification can be divided into three subcategories: the optimization approach which makes optimal modification to the graph [108]; the randomized modification approach which conducts perturbations [109–111] and the greedy modification approach [104] which modifies the graph in a greedy way such that the privacy and data utility is preserved [102].

In clustering approaches similar nodes and edges of the graph with same structural properties are grouped together and then in the published graph the nodes are replaced by the groups. In graph modification techniques the topological structure of the graph is modified by adding or deleting nodes and/or edges. In a graph all nodes and edges are correlated and a change can spread across the whole network and each change represents loss of utility. The above anonymization methods, in addition to naive anonymization, can preserve privacy in social network data [93].

6.2 The k anonymity model

The most well known and one of the first methods presented to preserve data privacy is k-anonymity. K-anonymity method was initially defined by Samarati [112] and Sweeney [31] for protecting privacy of tabular data. Many proposed techniques are based on the k-anonymity model which aims to preserve privacy by making each individual indistinguishable from at least $k-1$ other individuals. Thus, an individual cannot be re-identified by an attacker with probability higher than $1/k$.

Later researchers adopted this technique also for protecting privacy in network and social data [113]. To apply k- anonymity on graphs, firstly it is important to identify the attributes which can be used and linked to external information to re-identify the individuals (quasi-identifier). Some data attributes can be used as quasi-identifiers, such as the degree of the nodes, the neighborhood etc.

The goal of the k-degree anonymity technique [108] is to protect against an adversary who has knowledge of the degree of some target nodes. If an adversary is able to identify a single node with the same degree in the anonymous graph, then is able to re-identify the node. This technique modifies the graph structure such that all nodes satisfy k-anonymity of their degrees, by modifying the edges. At the end, all nodes have at least $k-1$ other nodes with the same degree.

Instead of using the node degree, another quasi-identifier can be the neighborhood of a node. Zhou and Pei [104] proposed the k-neighborhood anonymity technique, considering 1-neighbourhood subgraph of the target node. This method protects from attackers with not only node degree as background knowledge, but also the neighborhood topology of a target node. However, the authors in [114] showed that the algorithm in [104] cannot handle attacks in which the adversaries have more than 1-neighborhood knowledge.

Structural information about a node can also be considered as a quasi-identifier. The k-automorphism algorithm [94] anonymizes a network and can guarantee k-anonymity even against an adversary who could know arbitrary hops in each user's neighborhood. In order to achieve this, the algorithm creates a supergraph of the original graph that satisfies the k-anonymity principle. In particular, a graph is k- automorphic if there are $k-1$ functions in the graph and for each node in the graph that an attacker cannot distinguish from the $k-1$ symmetric nodes. The k-automorphism technique protects the graph against neighborhood [104], degree-based [108] and subgraph attacks [105]. However, these techniques can compromise the privacy of the relationships between nodes, even if their identity is completely hidden.

Similar technique is the k-isomorphism anonymization approach [115]. In this method, the graph is partitioned in k subgraphs with the same number of nodes and all subgraphs become isomorphic by adding or deleting edges. The attackers cannot determine that two identities are connected by a path of certain length with probability more than $1/k$. This technique protects the graph from neighborhood attacks [104].

Other quasi-identifiers used are modelling more complex background knowledge of an attacker. For example, in the k-candidate anonymity [109] technique, a node is k-candidate anonymous with respect to query Q if there are at least $k-1$ other nodes in the graph that match query Q.

K- degree anonymity has been criticized, as it considers that an attacker pos-

sesses too little background knowledge. Models such as k-neighborhood and k-automorphism are more complex. These methods rely on their complexity, which however can prevent them from working effectively and efficiently on large network data.

However, most of these studies deal with simple graphs (unweighted, undirected, loopless). There has been an increasing interest in the analysis of weighted networks. Anonymizing a weighted social network is much more challenging than anonymizing simple graphs. Therefore, beyond the identity and relationship privacy, the privacy of edge weights also needs to be studied [98]. Edge weight anonymization is important since if an adversary re-identifies a node, more information will be revealed if the edge weights are not anonymized. This may be possible even when the network has been k-anonymized.

6.2.1 Beyond k-anonymity

A social network that has been k-anonymized can still be vulnerable against specific types of privacy leakage [116]. If for example all nodes in an k-anonymous group have similar attributes and are associated with certain sensitive information, an attacker can derive that sensitive information of the group [93].

Machanavajjhala et al. [117] introduced the concept of l-diversity for tabular data. Based on this concept, each k-anonymous class requires l different values for each attribute. This method can protect not only from identity disclosure, but also from attribute disclosure. The authors also introduced two attacks that can compromise privacy in a k-anonymity model: the homogeneity attack and the background knowledge attack. In the first attack, the adversary can identify the sensitive attributes of an instance in case the values lack diversity. In the second attack, the adversary can identify the sensitive attributes of an instance in case he has background knowledge, such as information about the behaviour of an individual. The l-diversity model ensures that sensitive attributes are diverse in the same class. Xiao and Tao [118] also provide proof that l-diversity guarantees stronger privacy preservation than k-anonymity.

However, Li et al. [99], went further and studied the vulnerabilities of the l-diversity model. They introduced the concept of t-closeness. This method requires that the distribution of sensitive attribute values within each class need to be close to the distribution of the attributes in the whole dataset. Skewness attacks or similarity attacks, in which the l-diversity approach is vulnerable, are confronted using the t-closeness approach [99]. In particular, a class satisfies t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution through the entire dataset is bounded by a certain threshold. This approach protects from both identity and attribute disclosure.

On the other hand, Chester and Srivastava [119] have shown that t-closeness cannot be clearly applied to social networks. The α-proximity approach they proposed can protect against attribute disclosure attacks. Their algorithm modifies a graph with labeled nodes by adding dummy edges so that it is α-proximal.

All above mentioned approaches were further adopted for social network anonymization of unstructured social network data [113].

6.3 Background knowledge of attackers

Different forms of external knowledge can compromise the privacy in social networks. An adversary may have direct information about some individuals and their relationships by participating in the network. Background information may be also gained through public sources. Modeling the capability of an attacker in graphs is more complex [92,102]. The adversaries rely on background knowledge to be able to distinguish nodes and learn the relationships between the individuals from a released anonymized graph [93]. If certain knowledge can uniquely identify some nodes in a graph and is known by an adversary, the privacy of these entities can be breached even if the data have been modified before publication [98]. Research on privacy preservation techniques on graphs has shown that simple anonymization methods do not work because in case an adversary possesses background knowledge, such as the network structure, can re-identity or gain more information about the nodes of the anonymized graph.

Backstrom et al. [3] described two types of attacks in an anonymized social network: active and passive attacks.

- In an active attack, an adversary creates new accounts and relationships in the original network and uses them in order to find the targets and their relations in the anonymized network.

- In a passive attack, an adversary can identify itself in the anonymized network and discover the identities and its relations with other identities. These attacks are based on the observation of small uniquely identifiable subgraphs.

Personal information that describes an individual and the structural information that describes how an individual is connected to other individuals in social network data can be used as background knowledge by an adversary to compromise privacy. Each individual has personal unique identifiers such as the home address (semi-identifier) or the sex (quasi-identifier). Several identifiers can be combined to potentially identify an individual.

Zhou et al [102] listed different types of background information. The node attributes, relationships between individuals, node degrees, neighborhoods, subgraphs and graph metrics such as betweenness, closeness and centrality, describe information that can be used as background knowledge. Graphs that are not associated with any attributes and the links are not labeled, the only background information that an adversary can possess is related to the structure of the graph. The structural information includes [102]:

- Node attributes [106] are a set of individual attributes that can uniquely link a node to an individual. Node attributes often are modeled as labels in a social network. These attributes are similar to quasi-identifiers in re-identification attacks on tabular data.

- Degree [108] is the number of direct social links or relationships of an individual. This information can be used to map a target in a network.

- Link relationships [100, 107] describe the connectivity between individuals. These links may have labels, such as the channels which the people use to

communicate. In a social network, for example, an adversary may acquire information about specific information between friends that only use email to contact, and try to use this background knowledge to identify the target in the network.

- Neighborhood [104, 105, 109] refers to a set of neighboring entities that have direct social relationships to an individual which they might also have mutual links.

- Subgraph [3, 94] refers to a set of relationships which the target individual is connected to. The subgraph is a subset of the entire graph. An attacker can identify an individual if he is aware of specific relationships.

- Network graph metrics [120] can implicitly reveal an individual. Graph metrics can be used as background knowledge from attackers to breach privacy. Such metrics are the betweenness, closeness, centrality, path length, etc. For example, the centrality can reveal the head of a group.

6.4 Preserving utility

Social networks need to be anonymized in order to preserve the privacy of the individuals involved in the network. Anonymized networks are used by different applications, for example to analyze the global structure or to analyze microstructures [102], leading to different anonymization methods.

However, the anonymization technique may affect the utility of the network. The utility, also called information loss, of the published anonymized graph data depends on the type of analysis that is performed on the anonymized network and can be measured by various metrics. Such metrics can be put into two categories with regards to whether the goal is to preserve the graph properties of the published graph or maintain high quality of the results of executing aggregate queries on the released graph [102].

Analyzing general graph properties is one of the most important applications. Researches have developed various metrics to indicate the characteristics and structure of a network [121]. Some graph properties that need to be preserved are the betweenness, closeness, shortest distance, centrality and path length. The betweenness measures the degree an individual lies between other individuals in the network in their shortest path. The closeness measures the degree an individual is near to all other individuals in the network directly or indirectly. The shortest distance between a node and all other nodes that can be reached from it. The centrality counts the number of relationships to other individuals in the network. The path length is the distances between pairs of nodes in the network. Other graph properties include the diameter, clustering co-efficient of networks and degree sequences [93, 102]. The clustering co-efficient is the degree to which graph nodes tend to cluster and degree distribution is the probability distribution of node degrees over the social network.

Aggregate network queries compute the aggregate on some paths or subgraphs which satisfy some given query conditions. For example, a user is interested to find the nearest neighbor of a medical doctor node to a teacher node in a social network. The aggregate query will return the average of the distance between these

two nodes [93]. Customer relationship management is another field where aggregate network queries are used [102].

Wu et al. [93] add one more category, the graph spectral properties. The spectrum of a graph is defined as the set of eigenvalues of the graph's adjacency matrix or other derived matrices, and can provide global measures for some graph characteristics.

Quantifying the information loss is very challenging in anonymizing social networks. The best utility is achieved when the released anonymized graph is isomorphic to the original graph. In case the nodes and edges of a graph are labeled, the measurement of aggregate queries is more useful. Information loss associated with the graph structure changes need to be taken into account. Measures that are based on the structure of the graph check the changes of the graph components that occured, such as the number of nodes and edges added or deleted. Moreover, changes in the node attributes should be considered when the information loss is measured. A more sophisticated measure of utility would consider the loss of structural properties in the anonymized data. For example, if all edges are removed then the utility is zero. If the anonymization cost is low, this means that few changes have been made to the original graph, resulting in higher overall utility [100].

6.5 Proposal

The analysis of weighted social networks have raised an increasing interest, as such models can be used for analyzing various social phenomena. The weights of edges can reflect affinity between two nodes or represent the communication cost between two individuals. In a social network, for example, the weight can be the frequency two individuals communicate to each other. This information may be considered private.

When a graph contains edge weights, there is an additional goal to protect the privacy of the weights. The proposed algorithm in this dissertation focuses on publishing social network data which allows useful analysis without disclosing sensitive information. A novel algorithm is developed which preserves the privacy of the individuals in a graph, applying the k-anonymity model based on the generalization method, and at the same time minimizing the information loss keeping the utility of the anonymized graph in high level for better analysis and exploitation.

K nodes with similar neighboring characteristics and the same structural properties are grouped into the same supernode and their relationships are hidden as they are grouped into superedges. Structural properties are based on the similarity of neighbors and the edge weights. The edge weights are modified and uncertainty about the existence of these relationships is introduced. The utility is measured in terms of general structural properties of weighted graphs such as the degree distribution, the edge weight distribution, the path length distribution and the volume distribution. The solution prevents identity, edge and edge weight disclosure and the graphs are weighted, undirected and may contain loops. The work is inspired by the work presented by Toivonen et al. [122] and next chapters of Part II originate from the work of Skarkala et al. [123].

The basic steps of the proposed algorithm are the following:

Step 1. Naive anonymization All the identifiers of the original graph are removed and replaced by temporary identities. At this step the weights of the

original graph remain the same.

Step 2. Node generalization Similar nodes are grouped together into the same equivalent class, which have the same structural properties. Next all classes are collapsed into single supernodes. Supernodes consist of at least k nodes of the original graph in order to achieve k-anonymity, and each node will not be indistinguishable from at least $k-1$ other nodes in the same supernode. Even an individual will not be able to identify itself in the supernode.

Step 3. Edge generalization After the super nodes have been created, the corresponding edges are also collapsed into superedges. The superedges will represent the edges that connect the nodes, which are included in the supernodes, in the original graph.

Step 4. Edge weight anonymity In this step the edge weight of the superedge is modified to preserve the generalized graph by edge weight disclosure.

Step 5. Publish anonymized network The final anonymized network is published for further analysis. Minimal information loss is achieved but at the same time the privacy of each individual is preserved.

6.6 Contribution

Most of the previous works on network anonymization only handles unweighted networks. As it comes to our knowledge, all the clustering-based methods that have been proposed for weighted graphs, do not anonymize the network based on the neighborhood structure and are concentrating mostly to preserve the network data from identity disclosure. Also, as we are aware, all the cluster-based methods that have already been proposed for weighted graphs have not considered neighborhood based attacks.

Similar to our work is the one presented by Liu and Yang [124]. Our approach differs as we propose a complete solution for edge weighted graphs which is based on the neighborhood similarity of the nodes. In comparison with work [124], which preserves the network only from identity and edge weight disclosure by grouping nodes with the same weight bags into supernodes, our approach considers also link disclosure and groups nodes with the same neighborhoods and the same structural properties based on the similarity of neighbors and the edge weights. Grouping nodes based on some structural properties such as degree or weight bag, reveals more information from which an adversary can infer if a targeted node is included within a supernode. On the other hand in our proposal, it is more difficult to terminate the presence of a node in a supernode, since the only thing that is revealed is the number of nodes included in a supernode and nothing more about their neighboring properties. The generalization-based approach we use gives a stronger protection against identity disclosure, as it includes degree anonymity, weight bag anonymity, as well as neighborhood anonymity as its special cases.

In addition to preventing identity disclosure in weighted networks, the proposed method can optionally prevent edge disclosure and edge weight disclosure. For both of them, the k-anonymization process usually introduces some uncertainty. The amount of this uncertainty in the anonymized graph can be quantified, and also increased if needed to achieve the required privacy level. Anonymization techniques modify data in some way to protect privacy, but this may affect the utility of the data. The utility or information loss of the published anonymized graph data de-

pends on its use. We use the total volume of (squared) edge weight changes as a proxy for more complex and application-dependent utilities, and the method specifically aims to minimize changes in edge weights while achieving k-anonymity.

In the proposed algorithm, we consider both active and passive attacks. Using generalization methods specific patterns of relationships will be hidden and neighborhoods with the same structural properties will be grouped in the same supernode, achieving at the same time the minimum information loss. By exploiting the k-anonymity model, the released network is isomorphic to the original one. Utility can be measured in terms of the shortest paths and their lengths, where the shortest path between two nodes is defined as the path with the minimum sum of weights. An un-weighted algorithm is used as a baseline to evaluate our approach in terms of privacy and utility using real world networks.

6.7 Organization of Part II

Part II is structured as follows. Chapter 7 addresses previous works on privacy preservation of graphs and networks. The anonymization approach is presented in Chapter 8. In this chapter the problem and key concepts are defined in Section 8.1. The proposed anonymization algorithm for weighted graphs is explained in detail in Section 8.2. Section 8.3 analyzes the evaluation of the experimental results. Possible security threats and their confrontation are discussed in Section 8.4. Finally, a brief summary of Part II is given in Chapter 9.

Chapter 7

Related work

Huge amounts of data being available in public has attracted the attention of data mining researchers. The need to extract useful information and knowledge from these data have forced the data mining community to develop efficient techniques which will allow the exploitation of the plethora of available information. However, effective data mining mechanisms which prevent the disclosure of private information in social network data face a variety of challenges that have been raised due to the amount of data and the sensitive information that might contain [125, 126]. Aggarwal [127] gives an introduction to social networks and the challenges of the analysis of large social network data are discussed in [128]. Frikken and Golle [129] first presented the definition of privacy preserving social network analysis. The concern about privacy issues is growing due to the public release of real world social networks or graphs [92, 93]. Privacy preservation of graphs and networks has also been studied by Liu et al [92], and the authors discuss different privacy preservation strategies for graph structures. Hay et al. [130] also discuss privacy preservation methods and survey privacy threats and attacks in networks. Preservation of privacy on social networks analysis has also been researched for criminal investigations [131].

Preibusch et al. [132] present a complete survey on privacy in social networks. Users in social networks have the ability to control and manage social contexts and often believe that a "private" profile will protect them from any threat [133]. However, the authors in [132] argue that the user's privacy may be exposed to threats by their contacts' privacy settings. Zheleva's et al [133] results show that by being a member of a group, a user's private attributes may be revealed by the group affiliations or friend's of the user. Bonneau and Preibusch [134] point out that *"privacy in social networks is dysfunctional in that there is significant variation in sites' privacy controls, data collection requirements, and legal privacy policies"*.

Over the years, extensive surveys have been presented on anonymization techniques [92, 102, 130]. Identity disclosure is a major privacy breach in social networks, and adversaries with specific background knowledge can reveal the identity of the individuals involved. The simple anonymization technique, known as naive anonymization, which removes the personal identification information has been proved ineffective in case an adversary has background information about the structure of a graph [3, 109]. This anonymization method does not guarantee identity and link privacy in simple graphs. Link disclosure happens when an adversary infers that two individuals have a relationship. If just the unique identifiers are replaced, by observing the structure it is easy for an adversary to identify that two target

nodes are connected. Link re-identification can occur even when nodes cannot be distinguished. Link re-identification was also researched by Ying and Wu [120] and they consider the case in which an adversary does not have any background information, by studying the probability of existence of a link. On the other hand, Korolova et al [135] consider that an adversary may gain background knowledge on the neighborhood of compromised individuals to attack on edge privacy and collect global information about the network.

Zhou et al. [102] categorize the anonymization techniques into two general approaches: clustering-based and graph modification. They also present different types of background knowledge such as the existence or absence of nodes and edges, node degrees, neighborhoods, subgraphs and graph metrics. Hay et al. [109] observed that structure properties, such as the degree of a node or the degree of a node's neighbor can make an individual more distinguishable, in case for example the degree is unique. This way the individual can be re-identified, even if there is no background knowledge of the original label [104, 108]. Some researches on analyzing privacy of graph data have shown that sensitive information in social network data can be de-anonymized. De-anonymization techniques [136] have shown that the social network structure can be exposed by analyzing the anonymous versions. Watanabe et al. [95] also present an analysis of privacy threats for anonymized social networks, and show the need of privacy and utility preservation at the same time. Joshi and Kuo [137] present different ways of privacy violations in social network data and metrics on how to measure security and privacy in online social networks. Ciriani et al. [138] surveyed the k- anonymity concept and its possible variations. K- anonymity has been studied widely. In the following, different techniques for network anonymization are reviewed.

7.1 Unweighted graphs

7.1.1 Clustering based techniques

7.1.1.1 Clustering based techniques for preventing identity disclosure

The generalization approaches are applied in many privacy preservation methods. In this approach nodes with similar structural properties are grouped together into super nodes, and edges are joined in superedges. A clustering-based anonymization approach for an unweighted and undirected graph was proposed by Campan and Truta [139]. In their proposal nodes with attributes are clustered to become indistinguishable, and only the number of edges between nodes within a cluster is revealed. This approach is similar to our proposal, but instead focuses on unweighted graphs.

Hay et al [105] proposed an edge generalization approach for unweighted graphs. Their k-anonymous method groups nodes within the same neighborhood into supernodes and edges into superedges. Privacy based on this method will be preserved, but due to many topological changes, the utility is not clear if it can be preserved. The number of nodes within supernodes and the number of edges can be revealed. Also, the authors do not impose any restriction on neighbourhood attacks.

Another clustering approach for undirected graphs is the $i-hop$ degree approach proposed by Thomson and Yao [140]. This method clusters nodes based on their

degree and a combination of a node and its neighbor's degree. Groups of k size are created and edges are added and/or deleted so that each node within the same cluster remains anonymous. The authors proposed the t-means algorithm, a constrained version of k-means, that limits the number of nodes within a cluster. Moreover, they proposed the union-split clustering algorithm, which joins nodes with two nearest clusters, and in case the size is more than $2k$ the clusters are split to k size. The authors consider as background knowledge an approach similar to vertex refinement queries for zero and one level neighborhoods.

Attribute disclosure has not been taken into consideration in many works. However, another anonymization approach proposed by Campan and Truta [106] considers undirected graphs where nodes are associated with attributes and edges are not labeled. Utility is optimized by using simultaneously the attribute and structural information. The clustering-based method clusters nodes which are indistinguishable based on their relationships and their attributes and protects privacy by satisfying k-anonymity both for quasi-identifier attributes and quasi-identifier structural attributes. They measure structural information loss by introducing a metric based on error probability and measure attribute information loss by adopting a generalization metric. This approach due to topological changes is not clear if it provides utility preservation.

7.1.1.2 Clustering based techniques for preventing edge disclosure

Link mining is closely related to privacy preservation in graph data [141]. Getoor and Diehl [141] presented a survey on link mining, portraying that link prediction is one fundamental problem, through which the existence of a link between two nodes is estimated by observing the links and node attributes. Link disclosure can occur even if identity disclosure is prevented and each node is k-anonymous. For example, if node a in the same group has an edge with every possible node b in the same group, an adversary can predict that there is a connection by a certain link between these two nodes, without knowing which node is who. Such privacy violation is easy to execute since users of social networks are explicitly connected together [132].

Bhagat et al. [142] consider the links in a social network as rich interaction graphs. They propose two anonymization techniques grouping nodes into classes, and point out that this method does not guarantee privacy. They require an additional condition based on which two nodes who share a neighbor must be included in different groups, in order to prevent edge disclosure. Each node cannot be connected with two or more nodes in the same group. This condition ensures that nodes are k anonymous and edge disclosure is bounded by $1/k$. The authors assume that an adversary might know a part of the nodes and links in the graph, and evaluate their method allowing queries.

Zheleva and Getoor [100] consider edge disclosure prevention in unweighted graphs. They propose a two step edge generalization approach, where nodes are not labeled but edges contain labels. The authors achieve k-anonymity to anonymize the nodes data and consider four edge modification cases: all edges are removed, a part of edges is removed, none of the edges is removed, or edges are clustered. Although the above proposed approaches would preserve privacy, it is not explicit their usefulness as many topological features may be lost in the anonymized graph.

7.1.2 Modification-based techniques

7.1.2.1 Modification-based techniques for preventing identity disclosure

Another privacy preservation approach is the modification of the graph, by adding or deleting edges. This approach preserves the degree of each node in the original graph by randomly switching a pair of existing edges and repeating the process k times. Re-identification attacks are confronted but the utility of data is decreased [110] due to randomization as many topological features are lost. Ying and Wu [110] propose the performance of few edge perturbations while preserving the spectrum of the original graph, which has close relations with topological properties such as diameter, long paths and bottlenecks etc, preserving both identity and link disclosure. Randomization methods for preserving graph spectral characteristics were also proposed by Ying and Wu [110]. They examine the eigenvector values of nodes to choose where edges are added, deleted or switched.

A k-candidate anonymity approach was presented by Hay et al. [109] which is based on similar neighborhoods, where nodes and edges do not contain attributes. In the proposed method, on a candidate set of each node random number of edge deletions and additions occur such that each set is automorphically equivalent with at least $k-1$ other nodes. The authors presented three types of queries: vertex refinement, subgraph and hub fingerprint queries on naive anonymized graphs. They showed that the privacy of the individuals can be violated by applying subgraph queries and if an adversary has background knowledge on the neighborhood of the target node. This technique however works for average node degrees which is not applicable to large scale social networks.

Liu and Terzi [108] presented k-degree anonymity in which for every node there are at least $k-1$ other nodes with the same degree. The authors assume that adversaries have background knowledge of the node's degrees/relationships and focus on preventing identity disclosure. Their method modifies the edges by increasing the degree of the nodes so that they become indistinguishable from at least $k-1$ other nodes, as they have the same degree, trying to preserve the structure of the original graph.

Zhou and Pei [104] proposed an anonymization method which generalizes the node labels and edges are added to create similar neighborhoods, until each node is k-anonymous. They assume adversaries know only the local neighborhood, 1-neighborhood, of the target node and they guarantee that an adversary with such information cannot identify any individual with confidence higher than $1/k$.

Tripathy and Panda [114] modified the technique proposed in [104], with labeled nodes. They create isomorphic neighborhoods by adding edges in order to prevent re-identification, assuming the adversary has background knowledge about the nodes within a finite number of hops from the target node. Their proposed method is based on adjacency matrix instead of DFS which was used in [104]. This approach results in less complexity time.

Zou et al. [94] proposed a k-automorphism algorithm which aims to construct a new graph so that for any subgraph around a node there are at least $k-1$ isomorphic subgraphs. They modify the graph by adding and deleting nodes and edges aiming to confront attacks that reveal the relationships between nodes and the structure of the target node. The authors assume that an adversary has background knowledge of the subgraph of the target node and the k-automorphism approach guarantees

privacy under any structural attacks, however it can be vulnerable to multiple other attacks. If such a subgraph is distinguishable in the anonymized graph, then the identity of the target node can be disclosed.

Wu et al. [143] add new nodes and edges to create k-automorphically equivalent nodes until every node has at least $k-1$ other nodes that are indistinguishable from it. The authors assume that an adversary might know the entire graph and the location of the target node. In case the graph is k-symmetric, an adversary will not be in position to identify the target node. The adversary is not realistic to be aware of all the neighborhoods of the target node, rather than just a part of the graph structure.

Yuan et.al. [144] focus on unweighted graphs with labeled nodes and one type of labels for edges. Their approach combines generalization and structure protection techniques such as micro data protection techniques by adding noise nodes or edges, and they define three levels of protection requirements based on the background knowledge of an attacker.

Clarkson et al. [101] transform a graph by adding edges such that each degree appears at least k times, aiming to prevent identity disclosure. They focus on unweighted and undirected graphs which do not contain self loops or multiple edges. They assume the attacker has certain prior background knowledge and use degree anonymization and structural cost to measure utility.

Chester et al. [145] transform a graph by adding nodes, in order to partition the degree sequence into subsequences of k length, such that at least $k-1$ other nodes have the same degree. They consider both labeled and unlabeled graphs, and they prove that on labeled graphs k-anonymization with a constant number of node additions is NP-complete.

7.1.2.2 Modification-based techniques for preventing edge disclosure

Ying et al [146] compared k-degree anonymity to different perturbation methods. In order to measure the level of anonymity they proposed the a-posteriori probabilities. From their experiments they concluded that the graph properties are preserved better for given levels of anonymity.

Link identification attacks were also studied by Zhang et al [147]. They propose degree based methods using edge deletion and edge-based swaps, for reducing the probability of edge existence. They assume that an adversary might be aware of the nodes degrees. This approach however does not consider node re-identification, which can occur even if the edge existence probability is small. For that purpose, the authors proposed the notion of t-confidence, to protect from such attacks. Edge anonymity is provided if the ratio of actual edges and possible edges between the equivalent classes is greater than a given threshold t.

Cheng et al. [115] considered as well link identification attacks. They proposed a k-isomorphism approach and showed that identity and link protection is achieved by anonymizing k pair subgraphs. For large subgraphs however the approach of finding frequent subgraphs can have high cost. Graph isomorphism problem which determines whether two graphs are isomorphic is NP hard [115].

7.2 Weighted graphs

Most of the existing literature on social network anonymization methods focus on preserving privacy on simple undirected and unweighted graphs. Many efficient algorithms have been proposed by researchers such as k-degree [108], k-isomorphism [115], k-symmetry [143] and k-automorphism [94]. However these studies deal with simple graphs only. In social networks though there are stronger or weaker relationships between individuals which are represented by edge weights and can vary. Weighted graphs provide additional information which is sensitive, so anonymizing these types of graphs is much more challenging, as the weight related information can be used by attackers to compromise privacy and result in identity disclosure [96].

7.2.1 Modification based techniques

Anonymization methods on weighted graphs were first studied in [111,148]. Authors consider the edge weights sensitive and they proposed a perturbation method which preserves the shortest paths between pairs of nodes. However, in their approach the anonymized weights are close to the original weights, which can lead to the discovery of sensitive weight values by potential adversaries.

The authors in [149] and [150] extend the work by Liu et al. [148]. They focus on preserving the shortest paths while the edge weights are being anonymized. Based on their proposal, they re-assign the edge weights preserving at the same time a linear property of the original graph. The authors discussed two linear properties in details: single source shortest paths and all shortest path pairs. However, if an adversary possesses node degrees as background knowledge, some sensitive information can be re-identified.

Liu et al. [98] proposed a perturbation approach to anonymize edge weight. The authors preserve linear properties such as the shortest paths, and concentrate on preserving weight privacy by applying a k-anonymous algorithm to modify the edge weights based on random walk and matrix analysis.

Li and Shen [151] propose two perturbation-based weight anonymization methods. They introduce two volume sequence perturbation algorithms to reconstruct a graph, one with graph transformation and the other with problem reduction. They modify the weights of the graph while considering the change of graph spectrum as a metric for information loss and use algebraic connectivity as a quantitative metric. They consider identity disclosure and assume the sum of weights as background knowledge in order to protect privacy from volume attacks.

Li and Shen [96] propose volume and histogram anonymization in order to prevent weight based attacks. They consider re-identification and modify edges and edge weights. This method causes though information loss on the statistical properties of the weighted graph.

Wang et al. [152] proposed an anonymization method to preserve the sensitive links between two nodes in a social network. They focus on k-anonymous path privacy and perturb the minimal number of edge weights to create at least k indistinguishable shortest paths, hiding sensitive information and the true paths are not revealed.

7.2.2 Clustering based techniques

Liu and Yang [124] propose a generalization anonymization method in weighted undirected graphs and focus on protecting privacy while maintaining utility. They propose the k-possible anonymity to protect graphs against weighted based attacks, and they also investigate identity and edge weight disclosure. The proposed method achieves k anonymity by grouping nodes with similar weights bags into supernodes and edges are generalized using weight intervals. They evaluate their method using weight related measurements such as edge weight, degree, volume and path length.

Compared to the current proposal, in [124] the authors focus only on identity and edge weight disclosure. Modifying edges and edge weights to anonymize weight bags can preserve privacy from weight based attacks, but the utility is reduced. Except the information loss in each edge weight, the weight related statistical properties information loss need to be measured. The structure of the graph is also affected when anonymizing weighted graphs, but also the two connecting nodes are affected by modifying the weight of an edge. In our proposal the nodes are grouped based on the similarity of neighbors and edge weights. Grouping nodes based on some structural properties such as degree or weight bag, reveals more information from which an adversary can infer if a targeted node is included within a supernode. On the other hand in our proposal, it is more difficult to determine the presence of a node in a supernode, since the only thing that is revealed is the number of nodes included in a supernode and nothing about their neighboring properties.

In Appendix B, details about the above mentioned works are presented in Table B.2.

Chapter 8

Anonymization methodology

8.1 Preliminaries

The privacy of individuals participating in social networks, especially when these networks are published, is necessary to be preserved by an anonymization method which at the same time minimizes the loss of information. In this section the background of the proposed anonymization method is defined. Social networks are modeled as graphs that are weighted, undirected and contain loops, i.e., self-edges (Figure 8.1).

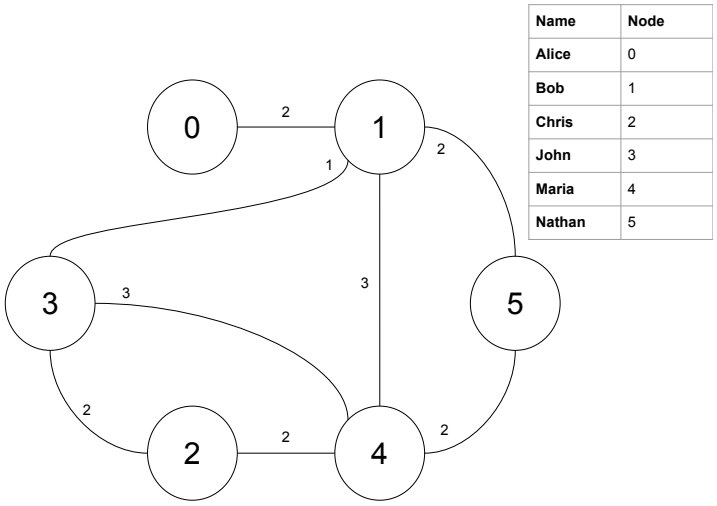


Figure 8.1: Network structure.

8.1.1 Problem definition

A weighted graph $G(V, E, W)$ is defined by a set of nodes V, a set of edges E and a positive weight function W. Given an edge $e_{i,j} = (i, j) \in E \subset V \times V$, we denote the weight by $w_{i,j} = W(i, j)$. As mentioned above, we consider graphs which are weighted, undirected and contain self edges. Similarly, the weight of a self-edge $e_{i,i}$ is denoted by $w_{i,i} = W(i, i)$. For notational convenience, in case two nodes are not connected the weight of their inexistent edge is defined as $w_{i,j} = 0$. Since the graph

is undirected the weights $w_{i,j}$ and $w_{j,i}$ are equal and the adjacency weight matrix is symmetric.

The following abstract problem is considered. Given a weighted graph G, produce a privacy-preserving version G' of it. The problem can be described as follows.

Problem. Given a weighted graph $G(V, E, W)$ and a parameter k, find the generalization that anonymizes G into a k- anonymous graph $G'(V', E', W')$, such that the information loss $IL(G, G')$ is minimized.

Different forms of external knowledge can compromise the privacy of individuals in social networks. An adversary may have direct information about some individuals and their relationships by participating in the network or from other sources. If certain knowledge can uniquely identify some nodes and edges between them in a network and is known by an adversary, the privacy of these individuals can be breached, even if the network data has been modified before publishing [96].

8.1.2 Preventing node identity disclosure

When social network data are anonymized in a privacy-preserving manner, one goal is to avoid identity disclosure. The k-anonymity model has been widely used in the literature for privacy preservation on graph data to achieve identity anonymization [105, 124, 139, 140].

K-anonymity, first defined in [112], is a strong property which captures the protection of data with respect to possible re-identification of the individuals to which the data refer. In the case of graph data, this anonymity property can be defined as follows.

Definition 1. *A graph is k-anonymous if every node in it is indistinguishable from at least $k-1$ other nodes.*

This property demands that every node in a graph is indistinguishably related to no fewer than k other nodes. Anonymization through k-anonymity, as described in Definition 1, is achieved either by adding edges or nodes in a graph or by grouping them together. In the present proposed method, we utilize the latter approach.

The basic idea is that original nodes in graph G are grouped into supernodes. Edges between the original nodes are replaced by superedges between the supernodes. A special case of superedges is the self-superedges which describe the connections between nodes included in the same supernode.

Definition 2. *A supernode sn_k in an anonymized graph G' represents a set of original nodes $sn_i, sn_j, \ldots$ in graph G.*

Definition 3. *A superedge $e_{i,j}$ in an anonymized graph G' represents a set of original edges in graph G that connect nodes which are included within supernodes sn_i and sn_j.*

A supernode represents all the original nodes contained within it and the relation between any two original nodes is described/approximated by the superedge between the supernodes that contain these original nodes. As a result, the only immediate information about original nodes is their supernode membership, and all other information is stored in the supernode.

While a supernode is simply a group of nodes, a superedge represents a hypothetical set of edges. This set may contain edges that do not exist in the original graph G. The weights may also differ from the original weights.

The graph data are generalized in order to produce a k-anonymous graph G'. To produce a k-anonymous graph, we need to group the original nodes of graph G into supernodes of size at least k and to assign superedges and superedge weights between them. We call such a graph a k-anonymity grouping of G (Figure 8.2).

Definition 4. *A k-anonymity grouping of a graph $G = (V, E, W)$ is a partitioning of the set V of nodes into supernodes sn_i such that $|sn_i| \geq k$, followed by a conjunction of the corresponding edges and weights into superedges e_{sn_i,sn_j}, and modified edge weights w_{sn_i,sn_j}, respectively.*

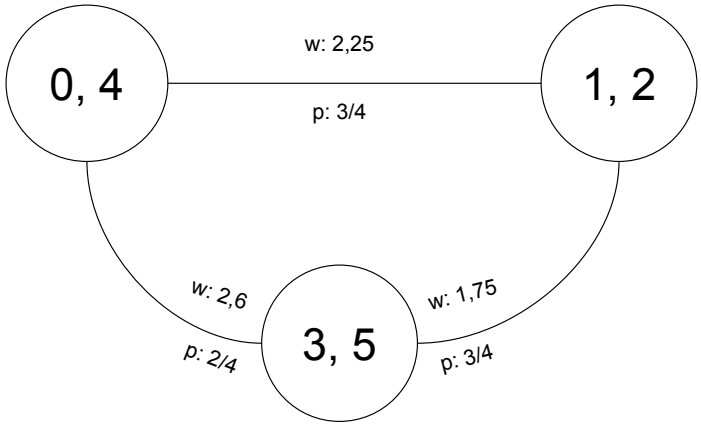


Figure 8.2: Anonymized Network[a].
[a] w denotes the edge weight and p denotes the probability of edge existence.

A k-anonymous graph G' consists of supernodes and superedges. A graph $G' = (V', E', W')$ is k-anonymous if the set V' of supernodes is a k-anonymity grouping of the set V of original nodes. This follows from the fact that a supernode represents all of its original nodes, so that at least k original nodes within a supernode become undistinguishable.

For analysis, the process of grouping nodes and edges has to be reversed to recover an approximate copy of the original graph. Original nodes are easily recovered

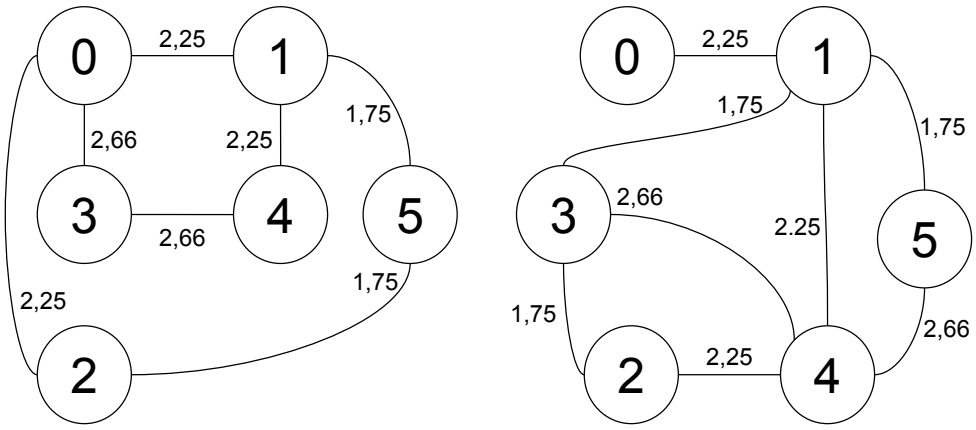


Figure 8.3: DeAnonymized Network[b].
[b] Two examples of the re-generated original graph.
Edges are randomly chosen based on the edge probability.

from supernodes, but edges and their weights may have changed. In particular, a superedge is associated with the number of true edges it represents but not their identities. To obtain an approximation of the original graph, that number of edges is materialized among all the possible edges a superedge represents. Since this results in some random changes of the network topology, it is a good practice to produce a number of alternative reconstructions, analyze all of them, and study the statistics over these graphs (Figure 8.3). In Section 8.3 we evaluate the proposed method using a number of statistical properties.

In the case of network data, k-anonymity grouping stands for a special case of the k-anonymity property. The grouping operation varies and depends on the property which will be used in order to choose the appropriate nodes that will portray a group. Such properties can be the degree, the weight bags or the neighbouring structure, which is used in our proposal so as to select at least k nodes that will form a group and each one will be indistinguishable among $k-1$ other nodes. Nodes with similar neighbouring properties, such as a sharing neighbour and similar edge weights are grouped together to achieve k-anonymity grouping or more specifically k-neighbourhood anonymity. More specifically, the k-anonymity grouping condition ensures that each group contains at least k nodes which implies that each node is indistinguishable from $k-1$ other nodes in the same group/supernode. A supernode generalizes and represents all its nodes, so they become structurally equivalent.

Thus, all the nodes within a supernode appear identical, and the supernode does not reveal any other information, other than the number of the nodes included. If an adversary would, for example, know the degree of a target node, the target node could still not be identified, because it can be a member of any supernode with a degree at least as high as its degree in the original graph. This means that the identity of individuals represented by a single supernode is masked preserving identity disclosure by any potential adversary. Therefore, if an anonymized graph G' is k-anonymous, an adversary cannot identify any node with confidence larger than $1/k$.

8.1.3 Preventing edge disclosure

In addition to the identity of individuals in social network data, information on relationships between the individuals may also be considered sensitive. In such cases, the anonymization of the network data should not just preserve the identity of the individuals, but also to protect the information on connections between them, i.e., to prevent edge and edge weight disclosure.

However, the k-anonymity as described above is not necessarily sufficient to prevent these two types of disclosure [98]. Assume, for example, that the network has been k-anonymized and that the adversary knows the degrees or the weight bags of two nodes. Now the adversary may be able to identify the supernodes that contain these two nodes, and also to infer even the existence of an edge in the original network. That is possible, because a superedge e_{sn_i,sn_j} does not only unveil a relationship between two supernodes, but also that the original nodes included in the supernodes are or can be connected.

Consequently, there is a need to prevent or at least to make it more difficult for an adversary to get definite information on the existence or non-existence of connections between nodes. One possibility to prevent this, is to avoid superedges that give absolute information about the existence of the original edges. Due to k-anonymity grouping, a superedge typically also represents edges that do not exist in the original graph, and therefore, this can be done easily using superedge probabilities.

Definition 5. *A superedge probability u_{sn_i,sn_j} describes the existence of edges $e_{i,j}$ between any pair of nodes included in supernodes sn_i and sn_j.*

The idea is that superedges are assigned probabilities, in order to prevent edge disclosure. Such a superedge probability can be defined as the percentage of original edges that are represented by a superedge. For example, in Figure 8.2 the probability of edge existence between supernodes $sn_{(0,4)}$ and $sn_{(1,2)}$ is 3/4. This means that there are actually 3 edges in the original graph G out of total 4 edges that the superedge represents in G'.

However, in some cases, for example, when the probability is close to 1, an adversary may still be able to infer the existence of an edge. To prevent such a situation all probabilities can be bound by a threshold. By keeping these probabilities below a given threshold p', an adversary can only infer the existence of an edge at most with confidence p', and edge disclosure can be guaranteed assuming that the adversary does not have any other relevant information about the original network. More specifically, except from the weight that describes the strength of each connection, each edge from the set E' is described by the probability of its existence bounded by a threshold. There are two ways to keep the probabilities low. One is to choose suitable supernodes and superedges. Another one is to artificially make superedges probabilities lower than p' in the output of the method. By introducing uncertainty about the existence of the original edge $e_{i,j}$ an adversary is unable to identify with 100% confidence that two nodes in G are actually connected.

8.1.4 Preventing edge weight disclosure

Edge weights give descriptive information about the relationships between two nodes in a graph. In social networks these weights can, for example, describe the strength of the connection between two individuals. Such information can be seen as sensitive, and if an adversary is aware of such weights, this information can be used to identify connected target nodes even in an anonymized graph. Especially, a unique pair of nodes in the anonymized graph with a specific edge weight leads to the re-identification of these nodes. Thus, protection of the edge weights, i.e., avoiding edge weight disclosure, should also be ensured before publishing the network data.

In the case of k-anonymity grouping, the weights assigned to the superedges are combined from the original edge weights $w_{i,j}$, since the edges $e_{i,j}$ between the original nodes are joined to the superedges. The weights of these superedges are called superedge weights w_{sn_i,sn_j}.

The combination of the original edge weights can be done in different ways and leads to the modification of the edge weight w_{sn_i,sn_j}. A superedge weight is defined as the average of the original edge weights, which results from Equation 8.1 where i' represents supernode sn_i and j' supernode sn_j.

$$w'_{i',j'} = \frac{\sum_{\{i,j\} \in i' \times j'} w_{i,j}}{|i'||j'|} \tag{8.1}$$

Most of the time, these superedge weights w_{sn_i,sn_j} differ from the weights $w_{i,j}$ of the original edges. Thus, the edge weight disclosure in those cases is prevented. However, in special occasions it may still be risky to publish these superedge weights as such.

For example, if $k = 2$ and $w'(i', j') = 0.25$ with edge probability $p(i', j') = 1$, publishing the superedge weight would immediately reveal that the only existing original edge between the nodes in those supernodes has exactly that weight. If the adversary knows all but one weight, that weight can be reverse-engineered. For more systematic and controlled protection, can be modified by a random component to add uncertainty of their real values.

On special occasions the weight w_{sn_i,sn_j} is the exact same weight as $w_{i,j}$. Possible approaches to attack this type of edge weight disclosure are to use a particular upper bound for the weight, or to define the weights as intervals. These approaches prevent the edge weight disclosure, but they decrease the utility of the anonymized graph G' to some extent.

8.1.5 Measuring information loss

K-anonymity grouping of nodes and combining original edges and their weights as superedges and superedge weights give us a privacy-preserving anonymization G' of an original graph G, and thus, solve the first part of our problem. However, we need to consider how much information is changed or lost in the process of network anonymization, as the generalization of the original graph can result in information loss which decreases the utility of the anonymized graph G'.

The information loss and data utility of weighted graphs, that are anonymized based on the concept of k-anonymity grouping, can be measured using different metrics. Our goal is to achieve k-anonymity with minimal information and use a metric that minimizes the information loss, and thus, guarantee high utility of the generalised graph. In order to achieve privacy-preserving anonymization, and at the same time increase the utility of the resulting graph G' is to make the anonymization by using an efficient information loss metric. One way to preserve the utility of the anonymized graph is to use the edge weight dissimilarity of the original graph G and the anonymized graph G'. This metric is presented in Equation 8.2.

$$IL(G, G') = \sum_{e_{i,j}} |W(i,j) - W'(i',j')|^2 \tag{8.2}$$

It can be shown that the sum in Equation 8.2 is minimized when the weight of each superedge is the average of the original edge weights [122]. Therefore, this average is used to modify the edge weights but at the same time preserve the utility of the released graph. The resulting weight is given by Equation 8.1 .

Our goal is to minimize the information loss that occurs when two nodes are possible candidates for grouping into the same supernode. By using this formula to narrow the information loss, we evaluate in Section 8.3 the reflection of this measure in the anonymized graph G' by comparing four different structural properties.

8.2 k-anonymization algorithm

The main goals of the proposed algorithm are to prevent identity, edge and edge weight disclosure by an adversary which possesses background knowledge of the original graph. A social network is modeled as a weighted graph G. An individual involved in the network is represented by a node in the original graph, and the connections between them by edges. The strength of their relationships are described by edge weights.

Based on the k-anonymity model, the main purpose of the anonymization algorithm proposed, is to keep the identity of these individuals hidden, making each of them indistinguishable from at least $k-1$ other individuals. The basic idea is that the original nodes are grouped into supernodes, the edges between original nodes are replaced by superedges between supernodes and edges between nodes that are contained within a supenode are replaced by a self-superedge.

In order to obtain a k-anonymity grouping of graph G with low information loss, the original nodes of the graph are grouped based on the similarity and strength of their relationships to other nodes and not based on their direct connections. The proposed anonymization technique aims to make nodes indistinguishable as far as the neighborhood is concerned. If a node has identical neighborhood set with another node in the original graph (they share the same neighbor), these two nodes are merged into a supernode in the anonymized graph G'. Specifically, two nodes that share a neighbour (they are $2-hop$ neighbours) and have similar edge weights are grouped together into a supernode and each supernode includes at least k original nodes with the similar neighbourhood properties. Thus, each supernode includes at least k original nodes and the only immediate information about original nodes is their supernode membership, as all other information is stored in the supernode. Nodes within a supernode are thus indistinguishable, from which k-neighbourhood anonymity of the original nodes follows. If an anonymized graph G' is k-neighbourhood anonymous, an adversary cannot re-identify each node within the supernode with confidence larger than $1/k$.

The original edges that connect nodes in graph G are grouped and represented by superedges between supernodes in graph G'. The grouping is based upon the results of Equation 8.1.5, aiming to minimize the information loss. The weight of a superedge connecting two supernodes is the average weight of the edges that connect the supernodes in the original graph. K-anonymity is not sufficient to prevent edge disclosure. An adversary may be able to identify that a node from the original graph is contained in a supernode with confidence $1/k$, and by observing the superedge may be able to infer the existence of an edge in the original graph. A superedge may represent edges that do not exist in the original graph G [114]. Using the generation of these in-existent edges, we introduce for each superedge and self-superedge the concept of uncertainty in order to confront edge disclosure. The idea is that not only the average weight of the original edges is assigned to superedges, but also the probabilities of the edge existence between the nodes included in a supernode. These probabilities stand for the percentage of the original edges (represented by the superedge) that actually exist. By inserting uncertainty about the existence or absence of edges, an adversary is unable to infer if two nodes are actually connected. An upper bound p' for this probability is given as a parameter. Now, the adversary can only infer the existence of an edge with confidence not larger than p'.

8.2.1 Anonymization algorithm

The basic steps of the proposed anonymization technique for social networks are following.

Step 1: Naive anonymization All the identifiers of the original graph G are removed and replaced by temporary identities. For example in Figure 8.1, the names of the individuals participating in the network are transformed into numbers, e.g. Alice is represented by number 0. By applying only this step, an adversary who does not possess any prior knowledge on G cannot re-identify any targeted node.

Step 2: Node anonymity Nodes with similar neighbourhood properties are grouped together into the same supernode and their corresponding edges into the same superedge, while information loss is minimized. In particular, two nodes that share a neighbour, which means their distance is exactly two hops, are merged into a supernode. Supernodes consist of at least k nodes of the original graph in order to achieve k-anonymity, and each node will be indistinguishable from at least $k-1$ other nodes in the same supernode.

For example in Figure 8.1, the neighborhood set of nodes 1 consists of nodes {0, 3, 4, 5}. These nodes are possible candidates for grouping, since node 1 is their common neighbor.

K-anonymity grouping indicates that each supernode should include at least k original nodes. The original edges that connect nodes in graph G are grouped and represented by superedges between supernodes in graph G'. In Figure 8.2, edge $e'_{(0,4),(1,2)}$ represents all the original edges that connect nodes 0, 1, 2 and 4.

Step 3: Edge weight anonymity The weights assigned to the superedges are combined from the original edge weights $w_{i,j}$, since the edges $e_{i,j}$ between the original nodes are joined to the superedges. The weights of these edges result from Equation 8.1 minimizing the information loss at the same time. The weight of a superedge connecting two supernodes is the average weight of the edges that connect the supernodes in the original graph. Therefore, in Figure 8.2 the weight $w'_{(0,4),(1,2)}$ is equal to the average weight of $e_{0,1}$, $e_{2,4}$ and $e_{1,4}$ edges.

In the anonymized graph in most cases the edge weight w' is different from the edge weight w, yet there is a chance that $w' = w$. In these cases, we consider an upper bound or weight interval in order to preserve the anonymized graph from edge weight disclosure. If original edge weights can be reconstructed from the anonymized graph, which means that the edge weight w' is similar to edge weight w, with error less than e (a user given parameter), then the superedge weight is changed by using a weight interval. This accessional action, however, may lead to the decrement of the utility of the anonymized graph.

Step 4: Edge anonymity Except from the edge weight w', the superedges are also described by the probability of edge existence p_{sn_i,sn_j}, which defines the percentage of original edges that are represented by a superedge. The probabilities of edge existence are computed for all superedges. If this probability is higher than p' (a user given parameter) then the probability is set to p'. Note that this anonymization strategy does not always protect from edge disclosure. In case the probability

is set close to 1, then an adversary may determine that all the nodes within two supernodes are also connected in the original graph. In this step the possibility of edge disclosure is eliminated as an adversary can only infer with confidence not larger than the probability that an edge between two original nodes exists.

Step 5: Publish anonymized network The final anonymized network G' is released for analysis.

8.2.2 Analysis of the anonymization algorithm

The *kAnonymous* algorithm (Algorithm 3) takes as input a weighted graph G and a parameter k and returns an anonymized graph G'. The original nodes are grouped in supernodes of at least k size and the edge weights of the superedges and superselfedges are set according to Equation 8.1. The algorithm works in a greedy mode since two original nodes (supernodes) and their edges (superedges/superselfedges) are grouped at a time until all the supernodes are containing at least k nodes, minimizing information loss.

Algorithm 3 *kAnonymous* Algorithm

Input: graph G, parameter k

Output: anonymized graph G'

1: **for** each original node n_i **do**

2: $$set \quad sn_i = \{n_i\}$$

3: **for** each original edge $e_{i,j}$ **do**

4: $$create \quad edge \quad e_{sn_i,sn_j}$$

5: **while** a node sn_i exists such that $|sn_i| < k$ **do**

6: select a random node sn_i such that $|sn_i| < k$

7: **for** nodes sn_j in $candidates(sn_i)$ **do**

8: $$IL_i = evaluate\_merger(sn_i, sn_j)$$

9: choose the node sn_j with the smallest IL_i

10: $merge(sn_i, sn_j)$

11: **end**

In the first two lines of the algorithm, each original node n_i in graph G is considered as a supernode in graph G'. In the next two lines, the corresponding edges e_{sn_i,sn_j} between the supernodes are created. Line 5 examines if there exists one or

more supernodes which include less than k nodes, i.e. if the network still needs to be anonymized. In case the anonymization is not yet complete, in Line 6 a random supernode is selected. In the following lines (7–10) this supernode is merged with another supernode. To implement different strategies for selecting the other supernode which will be merged, function *candidates* returns a list of possible options. Each of these candidates is considered for possible merger and finally the best candidate is chosen. Given a set of nodes obtained with function *candidates*, in Line 8 the information loss IL is computed for each one of them. The supernode sn_j with the smallest information loss is selected to be merged with supernode sn_i using the *merge* function.

Algorithm 4 *Candidates* function

```
1:  procedure candidates(sn_i)
2:      sn_k := 2−hop neighbors of sn_i
3:      if |sn_k| > 0 then
4:          sn_j := anonymity_cases (sn_k)
5:      else if |sn_k| = 0 then
6:          sn_k := neighbors of sn_i
7:          if |sn_k| > 0 then
8:              sn_j := anonymity_cases (sn_k)
9:          else if |NL| = 0 then
10:             set sn_k ≠ sn_i
11:             sn_j := anonymity_cases (sn_k)
12:     return sn_j
```

The *candidates* function (Algorithm 4) returns a set of candidate nodes with which node sn_i could be merged. Nodes with similar neighboring properties are grouped together, therefore the set of $2-hop$ neighbors of node sn_i is selected firstly. In case it is empty, the set of neighbors constitutes the next attempted candidate set, which in case is also empty is replaced by all the remaining supernodes that are different to sn_i and used as candidates. The last two candidate sets are used only in the special occasions in which a node does not have any $2-hop$ neighbors, so the set of neighbors can be used or in case the node is not connected to any other node, all the existing supernodes in the graph G' constitute the candidate set.

The *anonymity_cases* function (Algorithm 5) determines the candidate set based on three different versions of the proposed method. This function determines whether the candidate set will be randomly chosen, or will be chosen based on specific properties. In the first case a random candidate supernode sn_j is chosen. The *Anonymous* case returns a candidate sn_j which may already include k nodes. The *kAnonymous* case returns a candidate sn_j that has not been k-anonymized yet. In Section 8.3, we evaluate these three versions of our proposal since they give different trade-offs between speed and utility.

Algorithm 5 *Anonymity_cases* function

1: **procedure** *anonymize_cases*(sn_k)
2: **case** *Random*:
3: select random $sn_j \in sn_k$
4: **case** *Anonymous*:
5: $sn_j := sn_k$
6: **case** *kAnonymous*:
7: $sn_j := sn_k$ such that $|sn_k| < k$
8: **return** sn_j

The *evaluate_merger* function (Algorithm 6) merges a possible candidate with the selected supernode and computes the information loss, using Equation 8.1.5 that occurs in case supernode sn_i is merged with a possible candidate supernode sn_j. The weight w' is the average weight of the corresponding edges. The merging operation is revised and the resulting information loss is returned.

Algorithm 6 *Evaluate_merger* function

1: **procedure** *evaluate_merger*(sn_i, sn_j)
2: *merge*(sn_i, sn_j)
3: compute IL_j for grouping sn_i and sn_j (1)
4: undo the merge
5: **return** IL_j

Function *merge* (Algorithm 7) takes place after the selection of the supernode sn_j with the smallest information loss. A new supernode sn_{new} is created by the union of the best supernode sn_j and sn_i. For each node n included in the union of the neighbors of sn_i and sn_j, a new edge is created that connects this node n with the new supernode sn_{new}, computing at the same time the average weight of the corresponding edges. Every original edge that was attached to each one of the merged nodes is deleted.

The *kAnonymous* algorithm is repeated until all the supernodes in graph G' represent at least k nodes of graph G. In the following section, we evaluate three different modes of our proposal. In the first case, the *candidates* function returns a randomly chosen candidate sn_j which will be merged with the supernode sn_i (*Random*). In the second case, even if a supernode has already k nodes it is included in the set of candidates (*Anonymous*) while in the last case supernodes that are already k anonymous are not considered in the candidate set (*kAnonymous*).

Algorithm 7 *Merge* function

1: **procedure** $merge(sn_i, sn_j)$
2: create $sn_{new} := sn_i \cup sn_j$
3: $N_i :=$ neighbors of sn_i
4: $N_j :=$ neighbors of sn_j
5: **for each** node n in $N_i \cup N_j$ **do**
6: create edge $e_{n,sn_{new}}$
7: compute the weight $w'_{n,sn_{new}}$ (2)
8: delete edges e_{n,sn_i} and e_{n,sn_j}
9: delete nodes sn_i and sn_j
10: **end for**

An extended version of the *kAnonymous* algorithm (Algorithm 8) is presented in Appendix A.

8.3 Algorithm evaluation

In this section the anonymization method is evaluated using two real datasets, the Karate club [97] and Lesmis [153]. The first dataset contains 34 nodes and 78 edges and describes the network of friendships between the members of a karate club at a US university and the edges are indicating social interactions between two members. The second dataset contains 77 nodes and 254 edges and describes the network of co appearances of characters in Victor Hugo's novel "Les Miserables". Nodes represent characters and edges connect any pair of characters that appear in the same chapter of the book. The values on the edges are the number of such co appearances. Both networks are weighted and undirected. In our evaluation we did not use larger datasets for the reason that we wanted the first evaluation of our proposal to be more unambiguous. The usage of more complex datasets is one of the future plans.

A published anonymized network can be used for analysis by researchers. Our aim is to preserve the utility of the anonymized graph in high level for better analysis. For that purpose we measure the utility of the anonymized network that results after the execution of the proposed technique in terms of general structural properties of weighted graphs. We measure the degree, the volume distribution of all nodes in the graph, the edge weight distribution of all edges in the graph and the path length distribution between all pairs of nodes. The degree of a node in a network is the number of connections it has to other nodes. The degree distribution is the probability distribution of these degrees over the whole network. The volume is the sum of weights included in a weight bag, which is the multiset of the weights of the adjacent edges. The edge weight distribution is the distribution of weights assigned to all the edges of the network. The path length distribution is the distribution of lengths of shortest paths for all pairs of nodes.

The statistical properties were measured for the three versions of the proposed algorithm described in Section 8.2. To conduct the experiments two more algorithms were used to compare them with the *kAnonymous* algorithm. The randomized version chooses a random $2-hop$ neighbour of a node and groups the two nodes into the same supernode. The anonymous version finds for each $2-hop$ neighbour of a node, even the ones that are also contained to a k-anonymous supernode, the best grouping that minimizes the information loss. On the other hand, the k-anonymous version excludes from the candidate set the supernodes that are already k-anonymous, as it finds for each $2-hop$ neighbour that is not included in a k-anonymous supernode the group for minimum information loss.

For the evaluation of the utility of the anonymized graph, randomly chosen edges $e_{i,j}$ are sampled using the probabilities of edge existence. For example, if the probability of edge existence is 6/8 between two supenodes, it means that there are actually 6 edges in the original graph G out of the total 8 edges that the superedge represents. Thus, six edges between the nodes are randomly chosen in order to rebuild the original graph. From this random choice, also the degree, the path and the volume are affected. As shown in Figure 8.3 in Section 8.1, after the de-anonymization of graph G' different graphs can be obtained, either similar or not to the original graph.

The algorithms are implemented using Java programming language. The experiments were conducted on a computer system with 1,60GHz AMD E-350 Processor and 4GB RAM running the Windows 7 operating system.

8.3.1 Running performance

Tables 8.1, 8.2 and 8.3 presents the runtime of the three anonymization algorithms for the two real datasets and for different k parameters.

From the results, we conclude that the *kAnonymous* version for the Karate Club dataset requires little more time in the *Random* version but less than the *Anonymous* one, while the k parameter is getting larger. The running time of the *Random* version is logically smaller than the running time of the other two algorithms since it does not search for each possible candidate, but it picks one randomly. In both datasets, the *kAnonymous* version requires less time than the *Anonymous* one, while the k parameter is getting larger, since in the *Anonymous* version all possible candidates are examined. The results show that the increment of the k parameter has a small effect on the runtime of the *kAnonymous* algorithm for each dataset, since it does not examine if a candidate is already k-anonymous, which occurs for the second algorithm. For the Lesmis dataset the *kAnonymous* version follows the same pattern as the first dataset. Figure 8.4 and Figure 8.5 show a visual representation of the running times for both datasets.

Table 8.1: Running time for Karate club and Lesmis datasets ($k = 2$).

Algorithm	**Karate club**	**Lesmis**
Random	16 ms	82 ms
Anonymous	62 ms	323 ms
kAnonymous	47 ms	265 ms

Table 8.2: Running time for Karate club and Lesmis datasets ($k = 5$).

Algorithm	**Karate club**	**Lesmis**
Random	47 ms	156 ms
Anonymous	124 ms	422 ms
kAnonymous	47 ms	276 ms

Table 8.3: Running time for Karate club and Lesmis datasets ($k = 10$).

Algorithm	**Karate club**	**Lesmis**
Random	50 ms	171 ms
Anonymous	188 ms	687 ms
kAnonymous	70 ms	343 ms

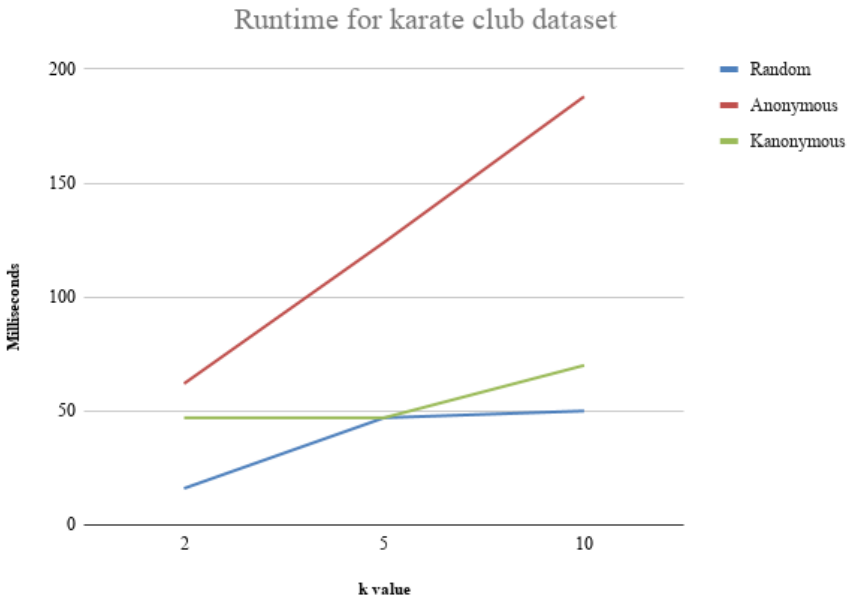


Figure 8.4: Running time for Karate club dataset

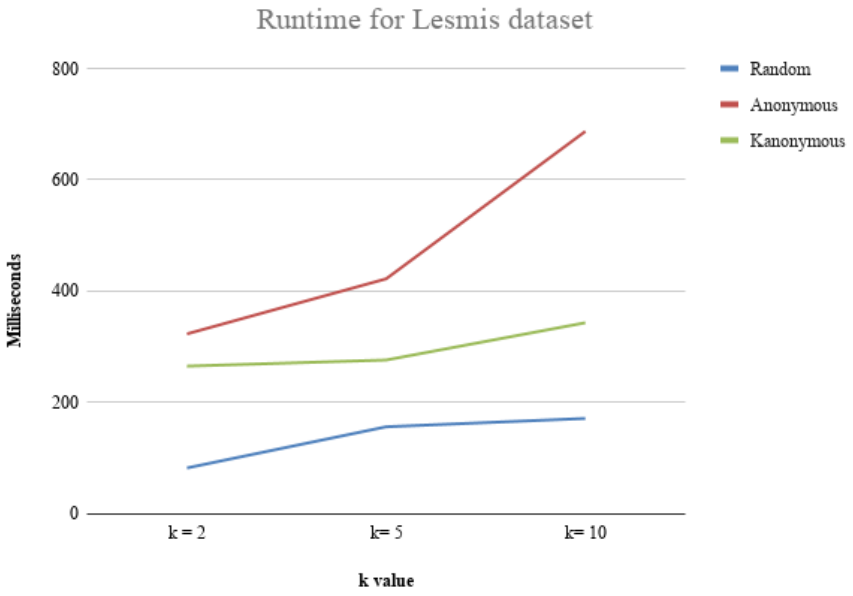


Figure 8.5: Running time for Lesmis dataset

8.3.2 Statistical properties

For measuring the utility of the anonymized graph, general structural properties were used such as the degree distribution, the edge weight distribution, the path length distribution and the volume distribution. The following figures present the statistical properties for both real datasets and for selected k parameters comparing the three anonymization versions with the original one.

In both datasets for $k = 2$ and $k = 5$, the degree distribution (Figure 8.6) of the three versions tend to be the same with the original one, while for $k = 10$ the

degree distribution of the *kAnonymous* version converges more to the original one. The *Random* and *Anonymous* version behave the same for most of the experiments conducted in the case of degree distribution.

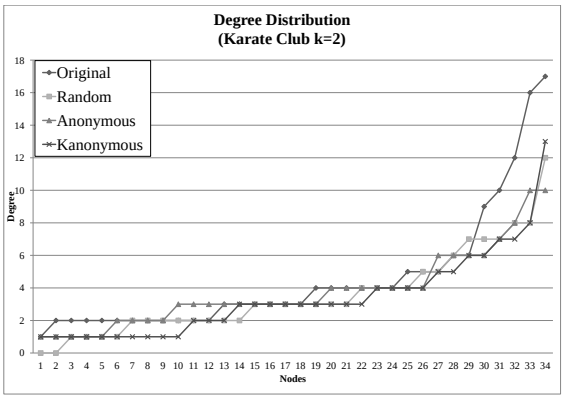
For the edge weight distribution (Figure 8.7) the inexistent edges in which the weight is zero were excluded, for better appearance of the results since they do not offer any further information. The *kAnonymous* version for the Lesmis dataset preserves better the edge weight of the anonymized graph than in the Karate club dataset for $k = 2$. While for $k = 5$, for both datasets, the edge weight distribution is maintained in high level by the *kAnonymous* version. On the other hand the *Anonymous* version and the *kAnonymous* one incline to the original edge weights for $k = 10$, in the Karate club dataset. In both datasets and for all k parameters, the edge weight distribution is maintained in high level by both the *kAnonymous* version and the *Anonymous* one.

The volume distribution (Figure 8.8) is preserved better by the *kAnonymous* version for both datasets for $k = 2$ and $k = 10$. Even the volume distribution of Lesmis dataset for the original graph and the *kAnonymous* one is almost the same for $k = 2$. For parameter $k = 5$, both *Anonymous* and *kAnonymous* versions preserve the volume of the nodes, having almost the same distribution as the original graph.

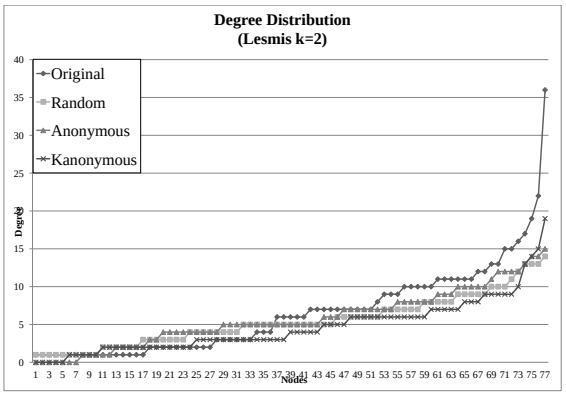
For $k = 2$ the path length (Figure 8.9) of the Karate club dataset is preserved better by the *kAnonymous* version while for the same k parameter in the Lesmis dataset the path length is maintained by the *Anonymous* version. For both datasets for $k = 5$ the *Anonymous* version performs better than the other two versions. On the other hand, the *kAnonymous* algorithm converse with the original graph's path length distribution for both datasets and for parameter $k = 10$.

While the k parameter is increasing the degree of the original nodes in the anonymized graph is decreasing in relation to the degree of the nodes in the original graph. This is reasonable, since the number of nodes included into supernodes depends on the k parameter, lowering at the same time the number of connections for each node. The volume of nodes is also affected since it is related to their degree, which is decreased, because it is related to the node's connections. The *kAnonymous* edge weights are kept in the same level as the original edge weights for $k = 2$, while the edge weights are decreasing for bigger k parameter. This decrement is raised because of the grouping of nodes that may cause more information loss due to the big number of the k parameter. The path length of the original graph is preserved more for bigger k, and even the original path length distribution of Lesmis dataset is almost the same as the *kAnonymous* path length for $k = 2$.

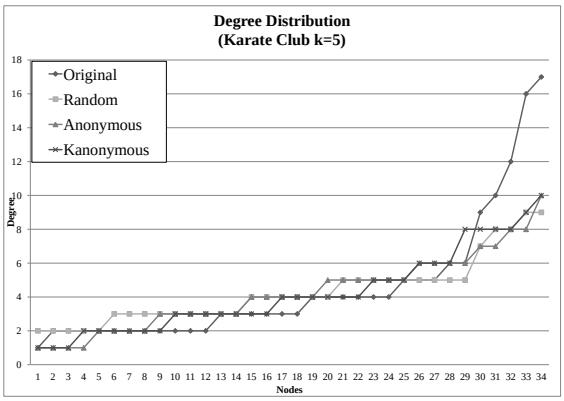
Our results demonstrate that the *kAnonymous* version of the graph preserves privacy and accurate results can be exported from the analysis of the anonymized graph. The original network can be recovered with little bias through aggregation on sampled graphs. From the results we conclude that both the *Anonymous* and *kAnonymous* versions can preserve in the same way three out of four statistical properties. Therefore, the examination or the pretermission of the already k-anonymous supernodes, does not affect the utility of the resulting anonymized graph.



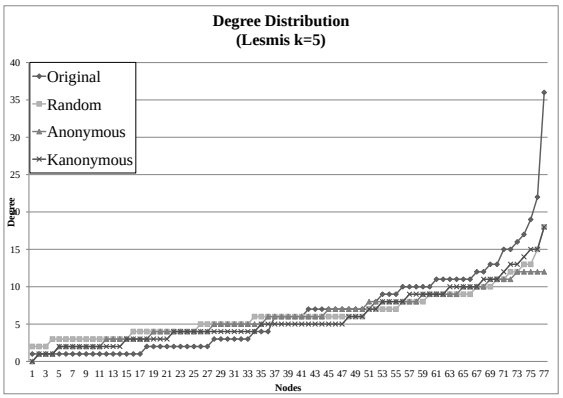
(a) Karate club dataset ($k = 2$)



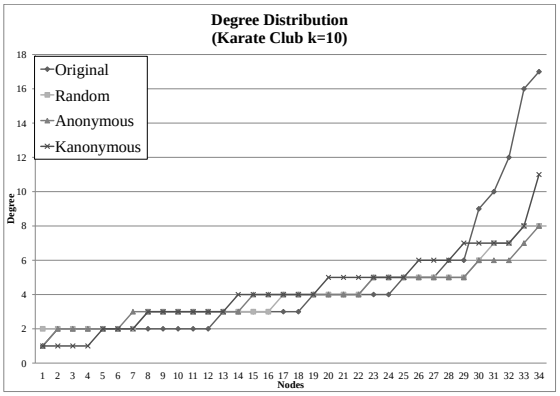
(b) Lesmis dataset ($k = 2$)



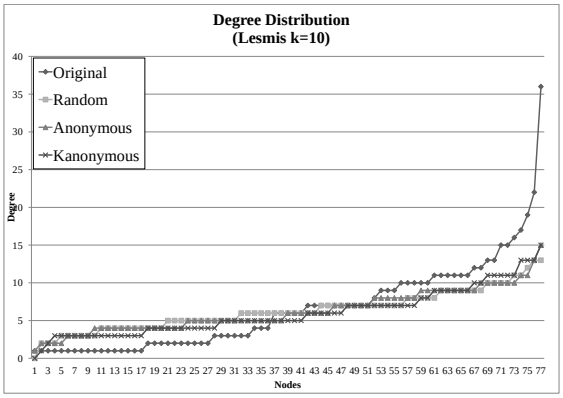
(c) Karate club dataset ($k = 5$)



(d) Lesmis dataset ($k = 5$)

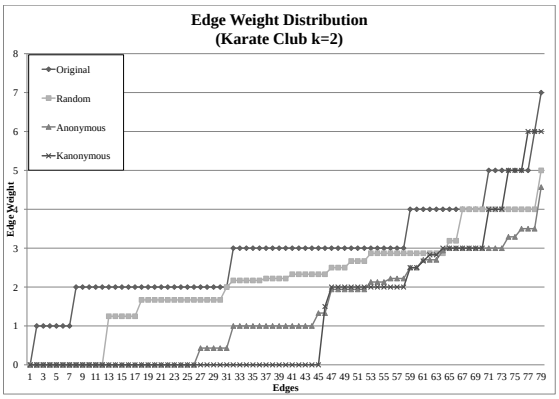


(e) Karate club dataset ($k = 10$)

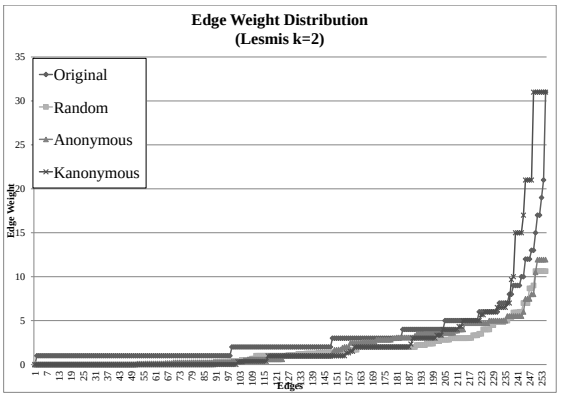


(f) Lesmis dataset ($k = 10$)

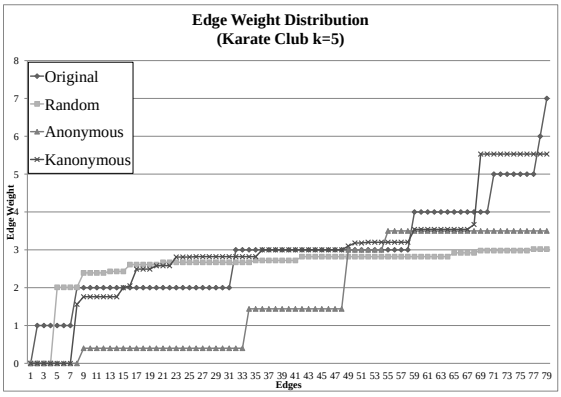
Figure 8.6: Degree distribution for Karate and Lesmis dataset



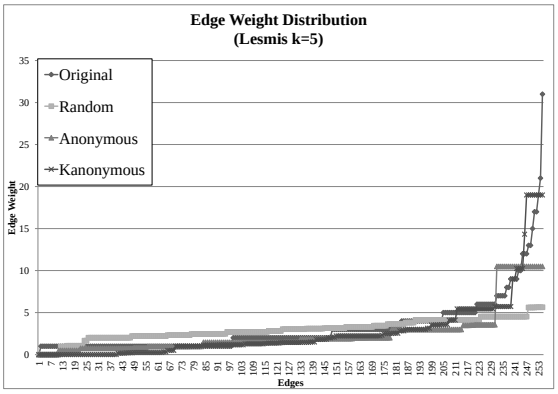
(a) Karate club dataset ($k = 2$)



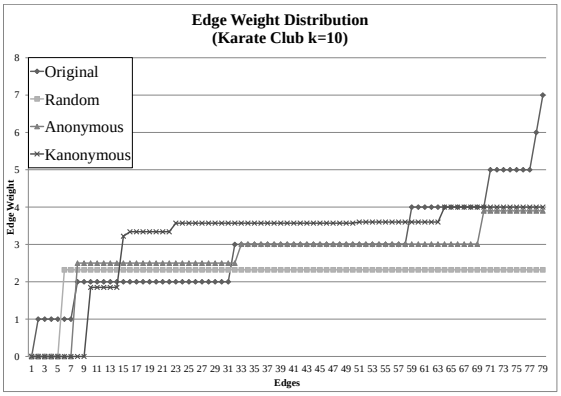
(b) Lesmis dataset ($k = 2$)



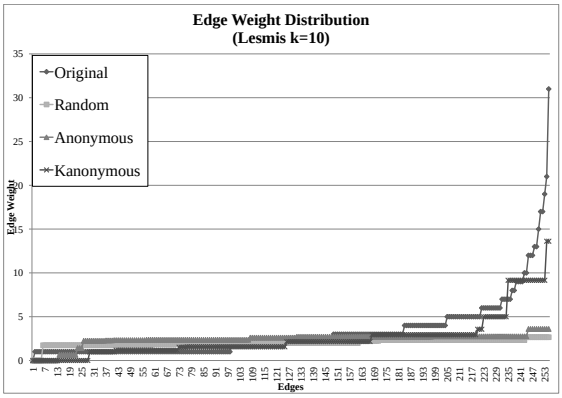
(c) Karate club dataset ($k = 5$)



(d) Lesmis dataset ($k = 5$)

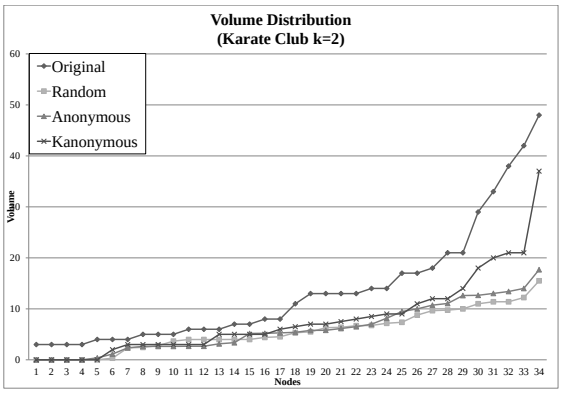


(e) Karate club dataset ($k = 10$)

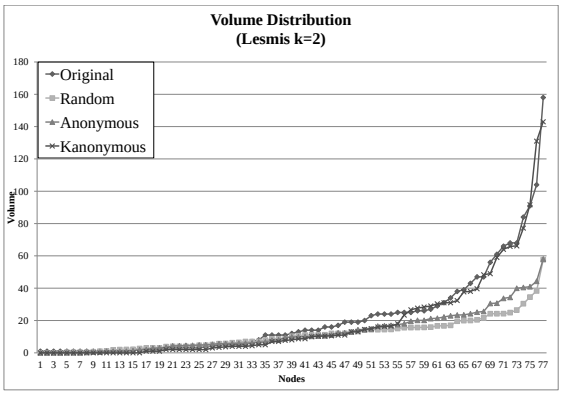


(f) Lesmis dataset ($k = 10$)

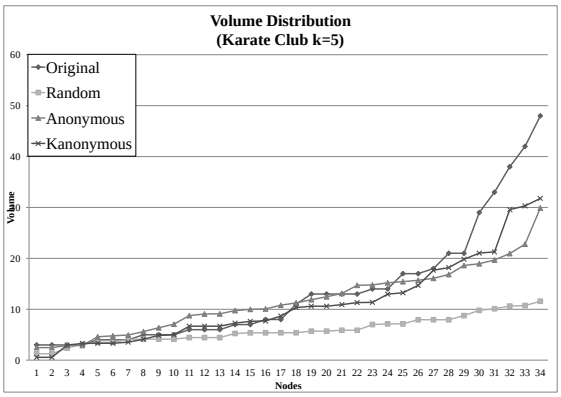
Figure 8.7: Edge weight distribution for Karate and Lesmis dataset



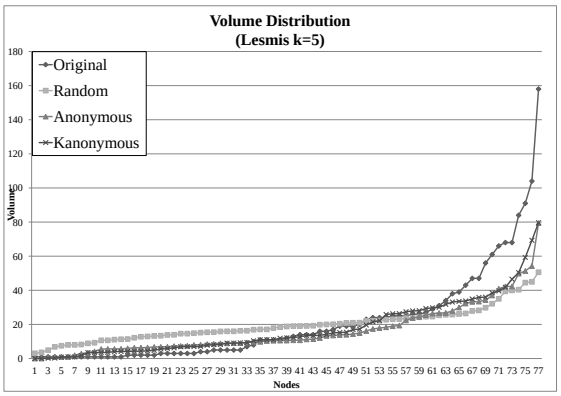
(a) Karate club dataset ($k = 2$)



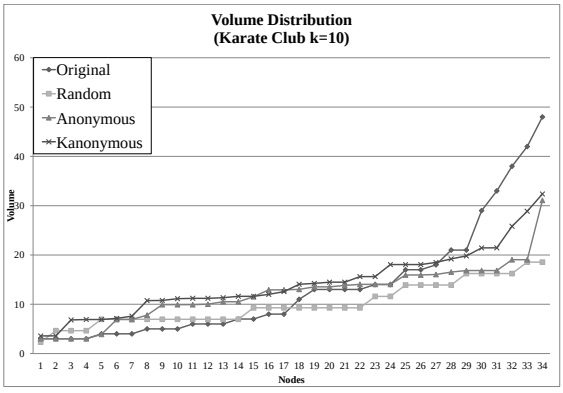
(b) Lesmis dataset ($k = 2$)



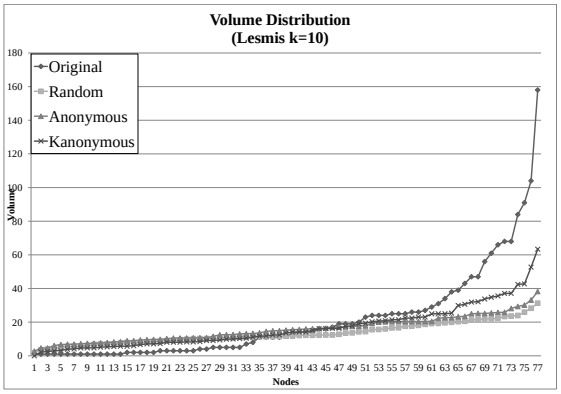
(c) Karate club dataset ($k = 5$)



(d) Lesmis dataset ($k = 5$)

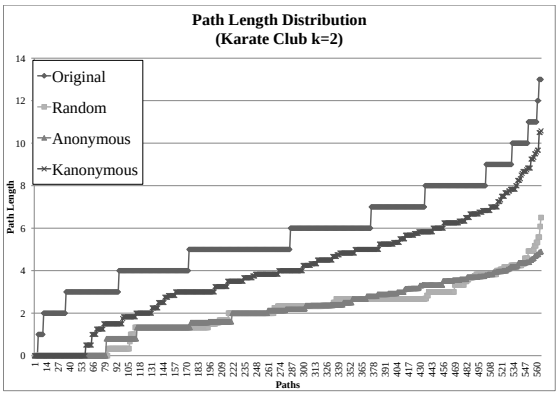


(e) Karate club dataset ($k = 10$)

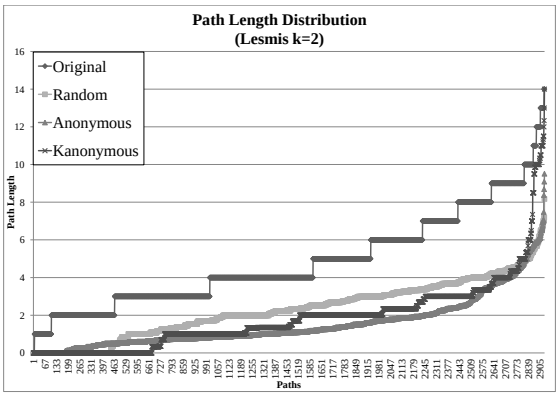


(f) Lesmis dataset ($k = 10$)

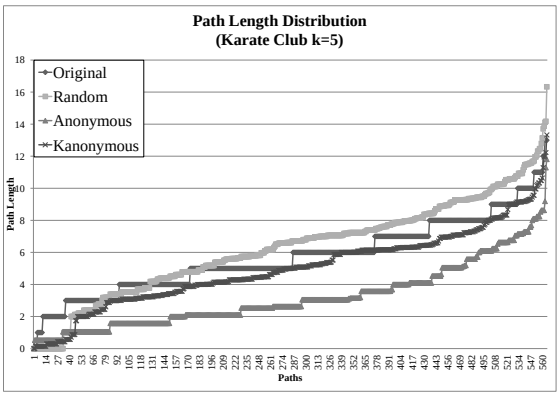
Figure 8.8: Volume distribution for Karate and Lesmis dataset



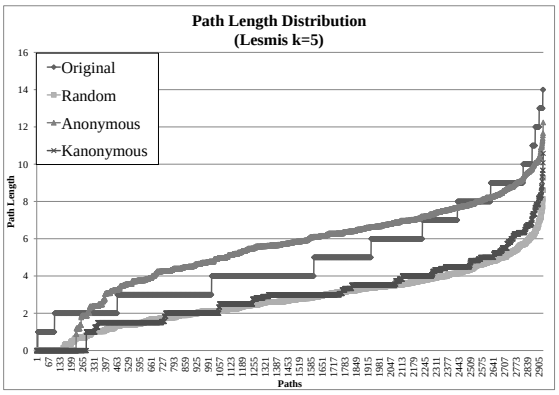
(a) Karate club dataset ($k = 2$)



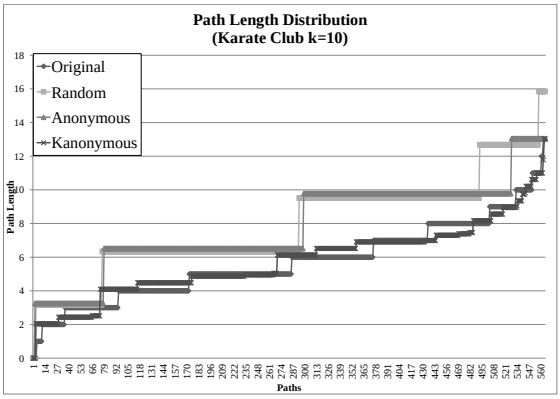
(b) Lesmis dataset ($k = 2$)



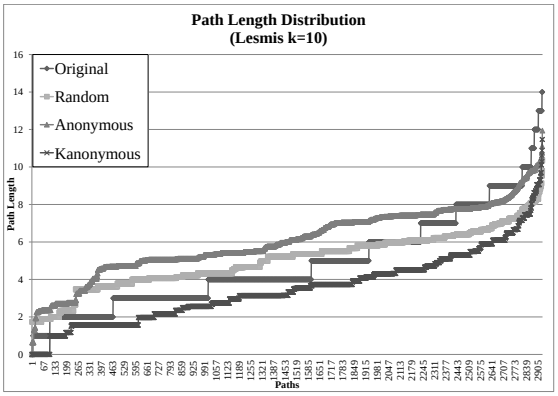
(c) Karate club dataset ($k = 5$)



(d) Lesmis dataset ($k = 5$)



(e) Karate club dataset ($k = 10$)



(f) Lesmis dataset ($k = 10$)

Figure 8.9: Path length distribution for Karate and Lesmis dataset

8.4 Threat model

Different forms of external knowledge can compromise the privacy in social networks. Background information regarding individuals and their relationships can be used by attackers to compromise the privacy of an anonymized network. There are three main types of privacy attacks in graph data: identity disclosure, link disclosure and content/attribute disclosure [99, 100]. In identity disclosure an adversary can determine an individual from an anonymized record/graph and can lead to attribute disclosure. In content/attribute disclosure an adversary can infer the attributes of an individual. In link disclosure an adversary can discover that two individuals have a specific relationship.

The background knowledge can be related to the neighborhood, degree, volume, subgraph etc. [102]. The degree of a node shows the number of connections/edges that this node has within the network. This information can be accessed by an adversary, if for example the attacker is aware of a specific number of node's connections, re-identifying that way the individual. Labels are used to describe the attributes of nodes. This information can be known by some adversaries which can help them re-identify data and compromise privacy. An attacker may also know specific relationships between some target individuals, and use this information to link the targets in the network. An attacker may also be aware of the neighborhood of some targets, and try to identify the targets in a network using this background information. Moreover, some graph metrics can also be used as background knowledge in order to identify individuals such as closeness, path length etc. [102].

The social network structure can be exposed by analyzing anonymized versions of networks. Most nodes in real social networks belong to a small uniquely identifiable subgraph [3]. Thus it is easy for an adversary to acquire subgraph background knowledge to conduct an attack. Subgraph attacks such that an adversary can learn the existence or absence of edges between specific targeted pairs of nodes from anonymized social networks are presented in [3]. An adversary can attack a network, either actively or passively, to identify a part of the network in order to gain background information. In an active attack the adversary creates new accounts and relationships in the original social network in such a way, so that these nodes and edges will be included within the published anonymized network. The attacker uses them in order to discover the identity of a target and its relations with other identities in the anonymized network. In a passive attack, the adversary tries to learn information about the nodes and edges after the anonymized network is published. The adversary does not create new nodes or edges for de-anonymization, but rather exchanges structural information with a small coalition of friends and uniquely identifies this coalition's subgraph, which enables those colluding friends to locate themselves.

By only changing identifying identities on a published social network, is not sufficient to preserve privacy, as an attacker can identify if there is a connection between a pair of nodes [3]. A sensitive relationship between individuals can be determined using different types of information such as edge existence, node attributes and structural properties [100]. Three types of queries, defined in [105], can assist an attacker to access information, i.e. subgraph queries, hub fingerprint queries, and vertex refinement queries. The vertex refinement queries describe the structure of the graph around a node, providing more information about the degree

of a node. Subgraph queries refer to the existence of a subgraph around a node, by counting the number of edges. Cannot exactly assert the existence or absence of edges in a network. A hub is a node with high degree. Hub fingerprint queries describe structural information about a node's neighborhood and connections to other hubs in the network [93]. If background knowledge is gained about the individuals by an attacker, even the publication of the anonymized network can cause privacy violation.

An adversary may have direct information about some individuals and their relationships by participating in the network or from other sources. If certain knowledge can uniquely identify some nodes in a graph and is known by an adversary, the privacy of these entities can be breached even if the data has been modified before publication [96]. Moreover, external databases can be accessed by attackers, which can gain knowledge on specific individuals in a social network. In order to prevent such attacks, the published network should be modified such that all identifiers are obscured but keeping the utility in high level.

Published social networks are vulnerable to neighborhood attacks. Two types of neighborhood attacks are the one-neighborhood attack and one-hop degree-based attack [154]. In the first attack, an adversary has background knowledge of the neighborhood and neighborhood relationships of a target node. In the second attack, an adversary has prior knowledge of the degree of the target node and the degrees of its one-hop neighbors. By using this knowledge, the attacker can identify the target node in the anonymized network.

Regarding the proposed anonymization technique, we assume that adversaries may have knowledge about some neighboring or weight-related properties of a targeting entity, such as the degree or the weight bag of a node. Assume that the adversary knows the degree of a target node. If an adversary has prior knowledge about the neighbors of a target and the relationship among the neighbors, the attacker may re-identify a target in a group of anonymized nodes that are all associated with some sensitive information. For example, if nodes within a supernode were grouped based on their similar degree.

Based on our proposal, the nodes are grouped in the same supernode if they share a neighbor. This however does not mean that they have the exact same degree. A given original node with a known degree can be a member of any supernode with a degree at least as high as the degree of the original node. Moreover, if an adversary knows the weight bag or the volume of a target node, and since the degree of the nodes included within a supernode may not be the same, this means that they do also have the same weight bag. However, all the nodes within a supernode appear identical, and the supernode does not reveal any other information, other than the number of the nodes included. After grouping, all these k nodes have identical degree, weight bag and volume. In this way, it is difficult for an adversary to infer which one of the k nodes is the target.

Link disclosure occurs when an adversary can determine whether a sensitive relationship between two given nodes exists. K-anonymity as defined above is not sufficient to prevent edge disclosure. Assume that the graph has been k-anonymized and that the adversary knows the degrees or the weight bags of the two nodes. Even in case an adversary may be able to identify the supernodes that contain the two targeting nodes, based on the superedge between them may be able to infer the existence of an edge in the original graph. Most of the works in the literature

introduce uncertainty about the existence of an edge that connects two nodes in a graph. By introducing uncertainty, using probabilities, about the edge existence between two targeting nodes an adversary can only assume that these two nodes are actually connected in the original graph, with confidence no larger than the probability u assigned to each superdge.

Regarding edge weight disclosure, assume an adversary knows the edge weight between two original nodes. Since, the weights of the edges are modified in the anonymized network by assigning the average of all edge weights that the superedge represents, the adversary is not able to identify the original edge weight. In case the average weight represents the exact original edge weight, then an upper bound can be used for preserving better the privacy of the edge weights, but the utility of the anonymized graph is not maintained in high level. It must be noted that in every experiment conducted all edge weights were modified, and did not represent the exact original one.

Chapter 9

Summary

Many concerns have been raised since sensitive social data are publicly released for research purposes. Social network data are analyzed by researchers, therefore the development of effective anonymization techniques became a necessity. The preservation of privacy of the individuals involved in a social network became the main concern in the social network analysis community.

Apart from the identities of social network participants, the relationships between these individuals need to be protected. Most existing works on privacy preservation over simple graphs deal with unweighted graphs. The proposed methods on unweighted graphs cannot be applied to graphs that contain more information about the individual relationships. Such information is the weight of the relationships between individuals, which is also considered sensitive.

In this part of the dissertation, a complete solution for effective and efficient anonymization is presented, focusing on weighted social network data. The basic preliminaries were introduced, in order to understand the background framework of the proposed technique. More specifically, a clustering-based k-neighborhood anonymization technique was presented. The method groups entities/nodes with the same neighboring properties into supernodes. Moreover, the connections of the nodes are grouped into superedges.

For measuring information loss, an effective metric was designed. Experimental results demonstrate that the proposed method strikes a balance between privacy and utility for real world weighted graphs. While preserving the utility of the weighted original graph, at the same time identity, edge and edge weight anonymity is achieved.

All clustering-based methods that have been proposed for weighted graphs in the literature, do not anonymize the network based on the neighborhood structure and have not considered neighborhood based attacks. They concentrate mainly in preserving the network data from identity disclosure. In addition to preventing identity disclosure, the proposed algorithm in Part II can prevent edge disclosure and edge weight disclosure, which are not considered in most studies proposed to the literature as it comes to our knowledge.

Part III

Conclusions and Future research

Chapter 10

Conclusions

Due to technological progress and the wide use of the Internet, voluminous data are stored in centralized or distributed databases. Data are exchanged daily for analysis and discovery of interesting hidden patterns and global information can be acquired by applying data mining techniques on statistical databases. Data analysis is important for a wide range of applications such as health care systems, businesses, insurances etc.

Nowadays, most systems are distributed and shared among several different parties. In most applications the data owners do not allow the disclosure of their own data. Databases often contain private data, and their disclosure when mining operations are applied could compromise the privacy. Therefore, privacy preserving data mining techniques have gained popularity and have become an important yet challenging field. This field joins two research areas in computer science, security and data mining.

Different privacy-preserving techniques focus on two main dimensions. The first one is the application of secure algorithms to protect the private data from being identified; data which can uniquely identify an individual. The second dimension is related to the final results which can be combined with external information and reveal private information about an individual, but not the identity.

There are two main privacy preservation methods for distributed data, cryptography and randomization. In Part I of this dissertation the first approach is utilized because of its security and accuracy of the final results. A privacy-preserving data mining technique was developed and presented for a distributed environment where databases can be horizontally or vertically partitioned. Both nominal and numeric attribute values are supported.

A trusted third party, the miner, acts as the data collector. The miner groups the received data and performs all secure operations to initialize and create the mining model. A secure version of the tree augmented Naive bayes was implemented for the purposes of the proposed method. However, the proposed protocol requires the participation of at least three parties who give as input their own private data. These three parties are not able to communicate with each other. The only workflow is between the miner and each one party. The Paillier cryptosystem, which exploits the homomorphic primitive, is used to encrypt all messages transmitted during the execution of the proposed protocol. Due to homomorphic primitive, privacy is preserved as the miner decrypts the messages received all at once.

Cryptography-based techniques do not modify or transform the original data.

Therefore, these techniques are considered the most appropriate approaches in terms of data accuracy, providing high quality of the final results. Simultaneously, all messages transmitted are examined for any type of modification. In particular, each message is concatenated with its summary, using the one-way hash function SHA-1. In the proposed protocol, both semi-honest and malicious adversaries are considered, and the threat analysis in Section 4.4 shows in detail how different types of attacks are confronted and the corresponding mechanisms used to preserve privacy.

Experiments were conducted to analyze the performance of all the main phases of the proposed protocol. The experiments took place for both partition types. The classifier was evaluated using three metrics: recall, precision and F1 score. The results showed that the proposed protocol is both effective and efficient. Its performance is affected when the amount of database attributes is increased. Using more advanced computer systems though the performance is almost four times improved compared to systems with less improved features.

In the literature, most of the proposed methods are not designed to support both partition types. Even more, these methods are mainly theoretical and have not been implemented to support real world applications. The contribution of the proposed protocol is significant as, to the best of our knowledge, none of the previously proposed techniques was designed and implemented for both horizontally and vertically partitioned databases while simultaneously providing accurate results and preserving privacy.

From the social networks perspective, data are being released publicly for analysis by different research fields such as psychology and sociology. Many concerns have been raised since these data can contain private information about the individuals involved and their identity might be disclosed. These concerns resulted in the development of effective anonymization techniques to preserve privacy.

Apart from the identities of social networks' entities, in many cases the connections between these entities are considered sensitive. These connections can reveal even more information if they are characterized by their strength, meaning their connection weights.

Clustering-based and modification methods are the two main categories of anonymizing social network data. In Part II of this dissertation we present an anonymization technique for effectively anonymizing social network data. The social networks considered are represented by weighted and undirected graphs. Specifically, a clustering-based k-anonymization technique is proposed which groups entities with the same neighboring properties. Similar nodes and their connections are grouped into supernodes and superedges, respectively. The proposed method prevents at the same time identity, edge and edge weight disclosure while preserving the utility of the original graph. An effective metric to measure information loss was introduced. Experiments conducted show that the utility is preserved for real world weighted graphs and a balance between privacy and utility is achieved.

Most existing methods on privacy preservation of simple graphs cannot be applied to weighted graphs. Clustering based approaches proposed in the literature do not consider neighborhood based attacks and they mainly focus on preventing identity disclosure. The proposed algorithm however provides mechanisms that prevent all three possible privacy breaches: identity, edge and edge weight disclosure.

The purpose of this dissertation is to present mechanisms which preserve privacy while data mining operations are applied to either tabular or graph data. Two meth-

ods were proposed: the first approach preserves privacy in tabular data distributed across multiple parties using cryptographic approaches; the second approach preserves privacy in network data, represented by weighted graphs. Both proposals have demonstrated that are efficient, performance related, and result in accurate outcomes without information loss.

10.1 Open issues

Defining privacy is challenging and several definitions have been proposed [12]. Individual's own privacy is subjective. Because of the absence of a standard definition it is hard to measure privacy. Most proposed metrics are defined and related to specific applications only.

Evaluating a privacy preserving data mining algorithm in terms of performance is not enough [23]. The scalability and efficiency of a privacy preserving data mining algorithm produce different results for different databases. A framework that allows the complete evaluation of a privacy preserving data mining algorithm considering different parameters is a necessity to be implemented and proceed to an extensive comparison of existing privacy preservation techniques in real world applications.

Systems that allow users to control their own data privacy need to develop the concept of personalized privacy. Personalized privacy though is challenging, as the user's idea of privacy does not correspond to their actions. Their concerns and actions can create a trade-off between privacy and utility, but also when the users are not aware of the privacy risks their actions can lead to private data disclosure. Solutions related to personalized privacy concept are yet to be implemented.

Cryptographic approaches can achieve privacy without compromising utility. These techniques however can lack efficiency for real world applications. Their development should be focused on preventing privacy of huge datasets and offer a scalable functionality which can be applied in the industry.

Background knowledge of adversaries is hard to define and model. Identifying the data that can be used for de-anonymization of public data that can be linked together is complicated. More realistic models that describe the background knowledge that is available to potential adversaries need to be developed.

Around the world many governments and public institutes are pressed to release data publicly, due to transparency requirements. By releasing more information, which can be obtained and analyzed by researchers, sensitive information may be exposed. Therefore, the new era of big and open data extends the exploration of the privacy preservation research field to future opportunities [116].

10.2 Future research

Research on privacy preserving data mining of distributed databases and networks has gone a long way and have been through several stages. The progress in the field however will continue in the upcoming years. As more privacy threats appear, new approaches will be developed and protocols will need to follow a common framework with specific definitions, principles and requirements.

The protocols presented in this dissertation can be extended in many directions. The proposed anonymization algorithm for privacy preservation of graphs can be evaluated using more complex social networks. The effectiveness also on other statistical graph properties can be investigated by carrying out experiments with larger datasets. In the future, extensive de-anonymization approaches on the resulting anonymized graph can prove the usefulness of the proposed algorithm in terms of privacy preservation and identity concealment through neighborhood based node grouping.

By comparing the proposed method for privacy preservation of distributed databases with ensemble methods such as random forest or gradient boosting machines could lead to the discovery of the most efficient and accurate algorithm. Furthermore, an extended comparison with El-Gamal's elliptic curve cryptosystem could be conducted in future research to achieve a balance between security and efficiency. The comparison of the computation cost of the main phases of the proposed protocol when either the Paillier or El-Gamal cryptosystem is applied could be examined in the future. The evaluation could be broadened by comparing the proposed method with previous schemes in the literature. Another interesting avenue for future research is the evaluation of the main procedures of the proposed protocol when more than three parties are connected to the miner. Conducting such experiments could prove the scalability and efficiency of the proposed protocol and how the number of participants affects the performance of the protocol. Finally, in the future, larger datasets and training sets from different real data sources could be exploited to evaluate the overall performance of the presented protocol.

An interesting future research, is the combination of the two proposed methodologies which can eventuate to a more efficient privacy preserving data mining technique, which can be used in real world applications by institutes and the industry. For example, a method which performs better can emerge from the combination of k-anonymity and homomorphic encryption. K- anonymous data can be combined with encrypted non-anonymous data and apply algebraic operations within groups instead of the whole dataset [155].

The goal of developing privacy preserving data mining techniques should be beyond the current status of creating basic methods that might not be applicable to real world applications. The future lies with methodologies that are implemented for institutes and businesses where privacy is essential and nowadays a public request.

Bibliography

[1] Dwipen Laskar and Geetachri Lachit. A review on "privacy preservation data mining (ppdm). *International Journal of Computer Applications Technology and Research*, 3:403–408, 07 2014.

[2] Oded Goldreich. Secure multi-party computation, 1998. Available online: `http://www.wisdom.weizmann.ac.il/~oded/PSX/prot.pdf` (accessed on 25 November 2020).

[3] Lars Backstrom, Cynthia Dwork, and Jon Kleinberg. Wherefore art thou r3579x? anonymized social networks, hidden patterns, and structural steganography. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 181—-190, New York, NY, USA, 2007. Association for Computing Machinery.

[4] Alpa Shah and Ravi Gulati. Privacy preserving data mining: Techniques, classification and implications - a survey. *International Journal of Computer Applications*, 137:40–46, 03 2016.

[5] Majid Malik, M. Ghazi, and Rashid Ali. Privacy preserving data mining techniques: Current scenario and future prospects. pages 26–32, 11 2012.

[6] Anastasiia Pika, Moe T. Wynn, Stephanus Budiono, Arthur H. M. ter Hofstede, Wil M. P. van der Aalst, and Hajo A. Reijers. Towards privacy-preserving process mining in healthcare. In *Business Process Management Workshops*, pages 483–495, Cham, 2019. Springer International Publishing.

[7] General data protection regulation (gdpr) – official legal text. Available online: `https://gdpr-info.eu/`(accessed on 25 November 2020).

[8] Krishnaram Kenthapadi, Ilya Mironov, and Abhradeep Guha Thakurta. Privacy-preserving data mining in industry. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, WSDM '19, page 840–841, New York, NY, USA, 2019. Association for Computing Machinery.

[9] Manish Sharma, Atul Chaudhary, Manish Mathuria, and Shalini Chaudhary. A review study on the privacy preserving data mining techniques and approaches. *International Journal of Computer Science and Telecommunications*, 4(9):42–46, 2013.

[10] Lei Xu, Chunxiao Jiang, Jian Wang, Jian Yuan, and Yong Ren. Information security in big data: Privacy and data mining. *IEEE Access*, 2:1149–1176, 2014.

[11] Charu C. Aggarwal and Philip S. Yu. A general survey of privacy-preserving data mining models and algorithms. In *Advances in Database Systems*, pages 11–52. Springer, Boston, MA, 2008.

[12] Ricardo Mendes and Joao P. Vilela. Privacy-Preserving Data Mining: Methods, Metrics, and Applications. *IEEE Access*, 5:10562–10582, jun 2017.

[13] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. Rappor. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security - CCS '14*, 2014.

[14] Apple Differential Privacy Team. Learning with privacy at scale differential. Available online: `https://machinelearning.apple.com/research/learning-with-privacy-at-scale`(accessed on 22 December 2020).

[15] Krishnaram Kenthapadi, Stuart Ambler, Liang Zhang, and Deepak Agarwal. Bringing salary transparency to the world. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 2017.

[16] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. Collecting telemetry data privately, 2017.

[17] Ljiljana Brankovic and Vladimir Estivill-castro. Privacy issues in knowledge discovery and data mining. In *In Proc. of Australian Institute of Computer Ethics Conference (AICEC99*, pages 89–99, 1999.

[18] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00, Dallas, TX, USA, 16–18 May 2000*, pages 439–450, New York, New York, USA, 2000. ACM Press.

[19] Yehuda Lindell and Benny Pinkas. Privacy preserving data mining. *Journal of Cryptology*, 15(3):177–206, 2003.

[20] Pingshui Wang, Tao Chen, and Zecheng Wang. Research on privacy preserving data mining. *Journal of Information Hiding and Privacy Protection*, 1(2):61–68, 2019.

[21] Elisa Bertino and R. Sandhu. Database security - concepts, approaches, and challenges. *Dependable and Secure Computing, IEEE Transactions on*, 2:2–19, 02 2005.

[22] Stan Matwin. *Privacy-Preserving Data Mining Techniques: Survey and Challenges*, pages 209–221. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[23] Elisa Bertino, Igor Nai Fovino, and Loredana Provenza. A framework for evaluating privacy preserving data mining algorithms*. *Data Min. Knowl. Discov.*, 11:121–154, 09 2005.

[24] Chris Clifton. Privacy preserving distributed data mining. Technical report, Department of Computer Sciences, West Lafayette, Indiana, United States, 2001. Available online: `https://www.cs.purdue.edu/homes/clifton/DistDM/CliftonDDM.pdf` (accessed on 25 November 2020).

[25] Chris Clifton and Don Marks. Security and Privacy Implications of Data Mining. *In ACM SIGMOD workshop on Research issues in data mining and knowledge discovery, Montreal, Canada, 2 June 1996*, pages 15–19, 1996.

[26] Avruti Srivastava. Comparative study of privacy preservation techniques in data mining. *International Journal of Computer Science and Information Technologies*, 5(4):7280–7287, 2014.

[27] Murat Kantarcioglu and Chris Clifton. Privacy preserving distributed mining of association rules on horizontally partitioned data. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1026–1037, sep 2004.

[28] Murat Kantarcioglu, Jaideep Vaidya, and Chris Clifton. Privacy preserving naive bayes classifier for horizontally partitioned data. In *IEEE ICDM workshop on Privacy Preserving Data Mining, Melbourne, Florida, United States, 19 - 22 November 2003*, pages 3–9, 2003.

[29] Rebecca Wright and Zhiqiang Yang. Privacy-preserving bayesian network structure computation on distributed heterogeneous data. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004*, KDD '04, page 713–718, New York, NY, USA, 2004. Association for Computing Machinery.

[30] Justin Zhan, Stan Matwin, and Li Wu Chang. *Privacy-Preserving Naive Bayesian Classification over Horizontally Partitioned Data*, pages 529–538. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[31] Latanya Sweeney. K-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, oct 2002.

[32] Vassilios S. Verykios, Elisa Bertino, Igor Nai Fovino, Loredana Parasiliti Provenza, Yucel Saygin, and Yannis Theodoridis. State-of-the-art in privacy preserving data mining. *SIGMOD Rec.*, 33(1):50–57, mar 2004.

[33] Dakshi Agrawal and Charu C. Aggarwal. On the design and quantification of privacy preserving data mining algorithms. In *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, Santa Barbara, CA, USA, 21–23 May 2001*, pages 247–255, New York, New York, USA, 2001. Association for Computing Machinery (ACM).

[34] Andrew Chi-Chih Yao. How to generate and exchange secrets. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science, Toronto, ON, Canada, 27–29 October 1986*, SFCS '86, pages 162—-167, Washington DC, USA, 1986. IEEE Computer Society.

[35] Yehuda Lindell and Benny Pinkas. Secure multiparty computation for privacy-preserving data mining. *IACR Cryptology ePrint Archive*, 2008:197, 11 2008.

[36] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Data Management Systems Series. Morgan Kaufmann Publishers, 2001.

[37] Charu Aggarwal and Philip Yu. A condensation approach to privacy preserving data mining. pages 183–199, 03 2004.

[38] Roberto J. Bayardo and Rakesh Agrawal. Data privacy through optimal k-anonymization. In *Proceedings of the 21st International Conference on Data Engineering*, ICDE '05, page 217–228, USA, 2005. IEEE Computer Society.

[39] Kristen LeFevre, David J. DeWitt, and Raghu Ramakrishnan. Incognito: Efficient full-domain k-anonymity. In *Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data*, SIGMOD '05, page 49–60, New York, NY, USA, 2005. Association for Computing Machinery.

[40] Chris Clifton, Murat Kantarcioglu, Jaideep Vaidya, Xiaodong Lin, and Michael Y. Zhu. Tools for privacy preserving distributed data mining. *SIGKDD Explor. Newsl.*, 4(2):28—-34, 2002.

[41] Nivedita Bairagi. A survey on privacy preserving data mining. *International Journal of Advanced Research in Computer Science*, 8:896–899, 2017.

[42] K. Saranya, K. Premalatha, and S. S. Rajasekar. A survey on privacy preserving data mining. In *2015 2nd International Conference on Electronics and Communication Systems (ICECS)*, pages 1740–1744, 2015.

[43] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, page 207–216, New York, NY, USA, 1993. Association for Computing Machinery.

[44] Aris Gkoulalas-Divanis and Vassilios S. Verykios. An overview of privacy preserving data mining. *XRDS*, 15(4), 2009.

[45] Elisa Bertino, Dan Lin, and Wei Jiang. *A Survey of Quantification of Privacy Preserving Data Mining Algorithms*, pages 183–205. Springer US, Boston, MA, 2008.

[46] Xinjun Qi and Mingkui Zong. An overview of privacy preserving data mining. *Procedia Environmental Sciences*, 12:1341–1347, 2012.

[47] Maria Eleni Skarkala, Manolis Maragoudakis, Stefanos Gritzalis, and Lilian Mitrou. Privacy preserving tree augmented naïve bayesian multi-party implementation on horizontally partitioned databases. In Steven Furnell, Costas Lambrinoudakis, and Günther Pernul, editors, *Trust, Privacy and Security in Digital Business*, pages 62–73, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[48] Maria Eleni Skarkala, Manolis Maragoudakis, Stefanos Gritzalis, and Lilian Mitrou. Pp-tan: a privacy preserving multi-party tree augmented naive bayes classifier. In *2020 5th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), Corfu, Greece, 25–27 September 2020*, pages 1–8, 2020.

[49] Maria Eleni Skarkala, Manolis Maragoudakis, Stefanos Gritzalis, and Lilian Mitrou. Ppdm-tan: A privacy-preserving multi-party classifier. *Computation*, 9(6), 2021.

[50] Olivier Baudron, Pierre-Alain Fouque, David Pointcheval, Jacques Stern, and Guillaume Poupard. Practical multi-candidate election system. In *Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing, Newport, RI, USA, 26–29 August 2001*, PODC '01, pages 274—-283, New York, NY, USA, 2001. Association for Computing Machinery.

[51] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29:131–163, 11 1997.

[52] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 223–238, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[53] Zhiqiang Yang, Sheng Zhong, and Rebecca N. Wright. *Privacy-Preserving Classification of Customer Data without Loss of Accuracy*, pages 92–102. 2005.

[54] Nan Zhang, Shengquan Wang, and Wei Zhao. A new scheme on privacy-preserving data classification. In *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining, Chicago, IL, USA, 21–24 August 2005*, KDD '05, pages 374—383, New York, NY, USA, 2005. Association for Computing Machinery.

[55] Y.A.A.S. Aldeen, Mazleena Salleh, and Mohammad A. Razzaque. A comprehensive review on privacy preserving data mining. *SpringerPlus*, 4(694), 2015.

[56] Murat Kantarcioglu, Jiashun Jin, and Chris Clifton. When do data mining results violate privacy? In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, WA, USA, 22–25 August 2004*, KDD '04, page 599–604, New York, NY, USA, 2004. Association for Computing Machinery.

[57] Simone Scardapane, Rosa Altilio, Valentina Ciccarelli, Aurelio Uncini, and Massimo Panella. *Privacy-Preserving Data Mining for Distributed Medical Scenarios*, pages 119–128. Springer International Publishing, Cham, Switzerland, 2018.

[58] Latanya Sweeney. Achieving k-anonymity privacy protection using generalization and suppression. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):571–588, 2002.

[59] Sheng Zhong, Zhiqiang Yang, and Rebecca N. Wright. Privacy-enhancing k-anonymization of customer data. In *Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '05, page 139–147, New York, NY, USA, 2005. Association for Computing Machinery.

[60] Mengdi Huai, Liusheng Huang, Wei Yang, Lu Li, and Mingyu Qi. Privacy-preserving naive bayes classification. In Songmao Zhang, Martin Wirsing, and Zili Zhang, editors, *Knowledge Science, Engineering and Management*, pages 627–638, Cham, Switzerland, 2015. Springer International Publishing.

[61] Kun Liu, Hillol Kargupta, and Jessica Ryan. Random projection - based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering*, 18:92–106, 02 2006.

[62] Jaideep Vaidya, Basit Shafiq, Anirban Basu, and Yuan Hong. Differentially private naive bayes classification. volume 1, pages 571–576, 11 2013.

[63] Jaideep Vaidya and Chris Clifton. Privacy preserving association rule mining in vertically partitioned data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Edmonton Alberta, Canada, 23–26 July 2002*, KDD '02, pages 639—-644, New York, NY, USA, 2002. Association for Computing Machinery.

[64] Wenliang Du and Zhijun Zhan. Using randomized response techniques for privacy-preserving data mining. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 505–510, New York, NY, USA, 2003. Association for Computing Machinery.

[65] Alexandre Evfimievski, Ramakrishnan Srikant, Rakesh Agrawal, and Johannes Gehrke. Privacy preserving mining of association rules. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '02, page 217–228, New York, NY, USA, 2002. Association for Computing Machinery.

[66] Hillol Kargupta, Souptik Datta, Q. Wang, and Krishnamoorthy Sivakumar. On the privacy preserving properties of random data perturbation techniques. pages 99–106, 12 2003.

[67] Peng Zhang, Yunhai Tong, Shiwei Tang, and Dongqing Yang. Privacy preserving naive bayes classification. In Xue Li, Shuliang Wang, and Zhao Yang Dong, editors, *Advanced Data Mining and Applications*, pages 744–752, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[68] Benny Pinkas. Cryptographic techniques for privacy-preserving data mining. *SIGKDD Explor. Newsl.*, 4(2):12–19, 2002.

[69] Jaideep Vaidya, Murat Kantarcioglu, and Chris Clifton. Privacy-preserving naive bayes classification. *The VLDB Journal*, 17(4):879–898, 2008.

[70] Xun Yi and Yanchun Zhang. Privacy-preserving naive bayes classification on distributed data via semi-trusted mixers. *Information Systems*, 34(3):371 – 380, 2009.

[71] Tamir Tassa. Secure mining of association rules in horizontally distributed databases. *IEEE Transactions on Knowledge and Data Engineering*, 26(4):970–983, 06 2011.

[72] Jaideep Vaidya and Chris Clifton. *Privacy Preserving Naïve Bayes Classifier for Vertically Partitioned Data*, pages 522–526. 2004.

[73] Jaideep Vaidya and Chris Clifton. Privacy-preserving k-means clustering over vertically partitioned data. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, page 206–215, New York, NY, USA, 2003. Association for Computing Machinery.

[74] Wenliang Du and Zhijun Zhan. Building decision tree classifier on private data. In *Proceedings of the IEEE International Conference on Privacy, Security and Data Mining - Volume 14*, CRPIT '14, page 1–8, AUS, 2002. Australian Computer Society, Inc.

[75] Weiwei Fang, Bingru Yang, Dingli Song, and Zhigang Tang. A new scheme on privacy-preserving distributed decision-tree mining. In *Proceedings of the 2009 First International Workshop on Education Technology and Computer Science - Volume 02*, ETCS '09, page 517–520, USA, 2009. IEEE Computer Society.

[76] W.-W Fang, B.-R Yang, J. Yang, and C.-S Zhou. Decision-tree model research based on privacy-preserving. 23:766–771, 12 2010.

[77] Jaideep Vaidya, Chris Clifton, Murat Kantarcioglu, and A. Scott Patterson. Privacy-preserving decision trees over vertically partitioned data. *ACM Trans. Knowl. Discov. Data*, 2(3), 2008.

[78] Bart Goethals, Sven Laur, Helger Lipmaa, and Taneli Mielikäinen. On private scalar product computation for privacy-preserving data mining. In Choon-sik Park and Seongtaek Chee, editors, *Information Security and Cryptology – ICISC 2004*, pages 104–120, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[79] B.N. Keshavamurthy, Mitesh Sharma, and Durga Toshniwal. Privacy preservation naïve bayes classification for a vertically distribution scenario using trusted third party. In *2010 International Conference on Advances in Recent Technologies in Communication and Computing, Kottayam, Kerala India, 16–17 October 2010*, pages 404–407, 2010.

[80] Chong-zhi Gao, Qiong Cheng, Pei He, Willy Susilo, and Jin Li. Privacy-preserving naive bayes classifiers secure against the substitution-then-comparison attack. *Information Sciences*, 444:72 – 88, 02 2018.

[81] Hwanjo Yu, Jaideep Vaidya, and Xiaoqian Jiang. Privacy-preserving svm classification on vertically partitioned data. In *Proceedings of the 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*, PAKDD'06, page 647–656, Berlin, Heidelberg, 2006. Springer-Verlag.

[82] Wei Jiang and Chris Clifton. A secure distributed framework for achieving ¡i¿k¡/i¿-anonymity. *The VLDB Journal*, 15(4):316–333, 2006.

[83] Madhuri Kumbhar and Reena Kharat. Privacy preserving mining of association rules on horizontally and vertically partitioned data: A review paper. In *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, pages 231–235, 12 2012.

[84] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, United States, 1 edition, 1997.

[85] C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. Inf. Theor.*, 14(3):462—-467, 1968.

[86] Michael G. Madden. On the classification performance of tan and general bayesian networks. *Knowledge-Based Systems*, 22(7):489 – 495, 2009. Artificial Intelligence 2008.

[87] Emmanouil Magkos, Manolis Maragoudakis, Vassilis Chrissikopoulos, and Stefanos Gritzalis. Accurate and large-scale privacy-preserving data mining using the election paradigm. *Data and Knowledge Engineering*, 68(11):1224 –1236, 2009.

[88] Hassan Takabi, Ehsan Hesamifard, and Mehdi Ghasemi. Preserving multi-party machine learning with homomorphic encryption. In *In 29th Annual Conference on Neural Information Processing Systems (NIPS), Barcelona, Spain, 5–10 December 2016*, 2016.

[89] Dheeru Dua and Casey Graff. Uci machine learning repository. Available online: `http://archive.ics.uci.edu/ml` (accessed on 25 November 2020).

[90] Istam Shadmanov and Kamola Shadmanova. Summarization of various security aspects and attacks in distributed systems: A review. *Advances in Computer Science*, 5:35–39, 01 2016.

[91] Ioanna Kantzavelou and Patel A. *Issues of attack in distributed systems - A Generic Attack Model*, pages 1–16. Springer US, Boston, MA, 1995.

[92] Kun Liu, Kamalika Das, Tyrone Grandison, and Hillol Kargupta. *Privacy-Preserving Data Analysis on Graphs and Social Networks*, pages 419–437. 12 2008.

[93] Xintao Wu, Xiaowei Ying, Kun Liu, and Lei Chen. *A Survey of Privacy-Preservation of Graphs and Social Networks*, pages 421–453. Springer US, Boston, MA, 2010.

[94] Lei Zou, Lei Chen, and M. Tamer Özsu. K-automorphism: A general framework for privacy preserving network publication. *Proc. VLDB Endow.*, 2(1):946–957, 2009.

[95] Chiemi Watanabe, Toshiyuki Amagasa, and Ling Liu. Privacy risks and countermeasures in publishing and mining social network data. In *7th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, pages 55–66, 2011.

[96] Yidong Li and Hong Shen. Anonymizing graphs against weight-based attacks. In *2010 IEEE International Conference on Data Mining Workshops*, pages 491–498, 2010.

[97] Wayne Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 33(4):452–473, 11 1976.

[98] Lian Liu, Cmida Lab, Jinze Liu, and Jun Zhang. Privacy preservation of affinities in social networks. In *Proceedings of the International conference on Information Systems*, pages 372–376, 01 2010.

[99] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *2007 IEEE 23rd International Conference on Data Engineering*, pages 106–115, 2007.

[100] Elena Zheleva and Lise Getoor. Preserving the privacy of sensitive relationships in graph data. In *Proceedings of the 1st ACM SIGKDD International Conference on Privacy, Security, and Trust in KDD*, PinKDD'07, page 153–171, Berlin, Heidelberg, 2007. Springer-Verlag.

[101] Kenneth Clarkson, Kun Liu, and Evimaria Terzi. *Toward Identity Anonymization in Social Networks*, pages 359–385. Springer New York, 08 2010.

[102] Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *SIGKDD Explor. Newsl.*, 10(2):12–22, 2008.

[103] Scott Coull, Fabian Monrose, Michael Reiter, and Michael Bailey. The challenges of effectively anonymizing network data. pages 230 – 236, 04 2009.

[104] Bin Zhou and Jian Pei. Preserving privacy in social networks against neighborhood attacks. In *Proceedings of the 2008 IEEE 24th International Conference on Data Engineering*, ICDE '08, page 506–515, USA, 2008. IEEE Computer Society.

[105] Michael Hay, Gerome Miklau, David Jensen, Don Towsley, and Philipp Weis. Resisting structural re-identification in anonymized social networks. *Proc. VLDB Endow.*, 1(1):102–114, 2008.

[106] Alina Campan and Traian Marius Truta. A clustering approach for data and structural anonymity in social networks. 2008.

[107] Graham Cormode, Divesh Srivastava, Ting Yu, and Qing Zhang. Anonymizing bipartite graph data using safe grouping. *VLDB J.*, 19:115–139, 02 2010.

[108] Kun Liu and Evimaria Terzi. Towards identity anonymization on graphs. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 93–106, New York, NY, USA, 2008. Association for Computing Machinery.

[109] Michael Hay, Gerome Miklau, David Jensen, Philipp Weis, and Siddharth Srivastava. Anonymizing social networks. Technical report, University of Massachusetts Amherst, 2013. Available online: `https://scholarworks.umass.edu/cgi/viewcontent.cgi?article=1175;context=cs_faculty_pubs` (accessed on 3 December 2020).

[110] Xiaowei Ying and Xintao Wu. Randomizing social networks: a spectrum preserving approach. pages 739–750, 04 2008.

[111] Lian Liu, Jie Wang, Jinze Liu, and Jun Zhang. Privacy preserving in social networks against sensitive edge disclosure. 2008.

[112] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowl. and Data Eng.*, 13(6):1010–1027, 2001.

[113] Ghazaleh Beigi and Huan Liu. A survey on privacy in social media: Identification, mitigation, and applications. *ACM/IMS Trans. Data Sci.*, 1(1), March 2020.

[114] B. K. Tripathy and G. K. Panda. A new approach to manage security against neighborhood attacks in social networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 264–269. IEEE Computer Society, 2010.

[115] James Cheng, Ada Wai-chee Fu, and Jia Liu. K-isomorphism: Privacy preserving network publication against structural attacks. In *Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data*, SIGMOD '10, pages 459—-470, New York, NY, USA, 2010. Association for Computing Machinery.

[116] Jordi Casas-Roma, Jordi Herrera-Joancomartí, and Vicenç Torra. A survey of graph-modification techniques for privacy-preserving on networks. *Artif. Intell. Rev.*, 47(3):341–366, 2017.

[117] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. ¡i¿l¡/i¿-diversity: Privacy beyond ¡i¿k¡/i¿-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1):3–es, 2007.

[118] Xiaokui Xiao and Yufei Tao. Personalized privacy preservation. In *Proceedings of the 2006 ACM SIGMOD International Conference on Management of Data*, SIGMOD '06, page 229–240, New York, NY, USA, 2006. Association for Computing Machinery.

[119] Sean Chester and Gautam Srivastava. Social network privacy for attribute disclosure attacks. In *2011 International Conference on Advances in Social Networks Analysis and Mining*, pages 445–449, 2011.

[120] Xiaowei Ying and Xintao Wu. On link privacy in randomizing social networks. pages 28–39, 01 2009.

[121] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, Jan 2007.

[122] Hannu Toivonen, Fang Zhou, Aleksi Hartikainen, and Atte Hinkka. Compression of weighted graphs. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, page 965–973, New York, NY, USA, 2011. Association for Computing Machinery.

[123] Maria E. Skarkala, Manolis Maragoudakis, Stefanos Gritzalis, Lilian Mitrou, Hannu Toivonen, and Pirjo Moen. Privacy preservation by k-anonymization

of weighted social networks. In *Proceedings of the 2012 International Conference on Advances in Social Networks Analysis and Mining (ASONAM 2012)*, ASONAM '12, page 423–428, USA, 2012. IEEE Computer Society.

[124] Xiangyu Liu and Xiaochun Yang. A generalization based approach for anonymizing weighted social network graphs. In *Proceedings of the 12th International Conference on Web-Age Information Management*, WAIM'11, pages 118—-130, Berlin, Heidelberg, 2011. Springer-Verlag.

[125] Jiawei Han and Gao J. Research challenges for data mining in science and engineering. In *Next Generation of Data Mining*, pages 3–27, 2008.

[126] Aparna S. Varde. Challenging research issues in data mining, databases and information retrieval. *SIGKDD Explor. Newsl.*, 11(1):49–52, 2009.

[127] Charu C. Aggarwal. *An Introduction to Social Network Data Analytics*, pages 1–15. Springer US, Boston, MA, 2011.

[128] Jon M. Kleinberg. Challenges in mining social network data: Processes, privacy, and paradoxes. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, page 4–5, New York, NY, USA, 2007. Association for Computing Machinery.

[129] Keith B. Frikken and Philippe Golle. Private social network analysis: How to assemble pieces of a graph privately. In *Proceedings of the 5th ACM Workshop on Privacy in Electronic Society*, WPES '06, page 89–98, New York, NY, USA, 2006. Association for Computing Machinery.

[130] Michael Hay, Kun Liu, Gerome Miklau, Jian Pei, and Evimaria Terzi. Privacy-aware data management in information networks. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, page 1201–1204, New York, NY, USA, 2011. Association for Computing Machinery.

[131] Florian Kerschbaum and Andreas Schaad. Privacy-preserving social network analysis for criminal investigations. In *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society*, WPES '08, page 9–14, New York, NY, USA, 2008. Association for Computing Machinery.

[132] Sören Preibusch, Bettina Hoser, Seda Gürses, and Bettina Berendt. Ubiquitous social networks: Opportunities and challenges for privacy-aware user modelling. *DIW Berlin, German Institute for Economic Research, Discussion Papers of DIW Berlin*, 01 2007.

[133] Elena Zheleva and Lise Getoor. To join or not to join: The illusion of privacy in social networks with mixed public and private user profiles. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, page 531–540, New York, NY, USA, 2009. Association for Computing Machinery.

[134] Joseph Bonneau and Sören Preibusch. The privacy jungle:on the market for data protection in social networks. In *Economics of Information Security and Privacy*, pages 121–167, Boston, MA, 2010. Springer US.

[135] Aleksandra Korolova, Rajeev Motwani, Shubha Nabar, and Ying Xu. Link privacy in social networks. pages 289–298, 04 2008.

[136] Arvind Narayanan and Vitaly Shmatikov. De-anonymizing social networks. *Proceedings - IEEE Symposium on Security and Privacy*, 04 2009.

[137] Prateek Joshi and C. C. Jay Kuo. Security and privacy in online social networks: A survey. In *2011 IEEE International Conference on Multimedia and Expo*, pages 1–6, 2011.

[138] Valentina Ciriani, SSF Vimercati, S. Foresti, and P. Samarati. *k-Anonymity*, volume 33, pages 323–353. 01 2007.

[139] Alina Campan and Traian Marius Truta. Data and structural k-anonymity in social networks. In Francesco Bonchi, Elena Ferrari, Wei Jiang, and Bradley Malin, editors, *Privacy, Security, and Trust in KDD*, pages 33–54, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[140] Brian Thompson and Danfeng Yao. The union-split algorithm and cluster-based anonymization of social networks. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, page 218–227, New York, NY, USA, 2009. Association for Computing Machinery.

[141] Lise Getoor and Christopher P. Diehl. Link mining: A survey. *SIGKDD Explor. Newsl.*, 7(2):3–12, 2005.

[142] Smriti Bhagat, Graham Cormode, Balachander Krishnamurthy, and Divesh Srivastava. Class-based graph anonymization for social network data. *Proc. VLDB Endow.*, 2(1):766—-777, 2009.

[143] Wentao Wu, Yanghua Xiao, Wei Wang, Zhenying He, and Zhihui Wang. K-symmetry model for identity anonymization in social networks. In *Proceedings of the 13th International Conference on Extending Database Technology*, EDBT '10, page 111–122, New York, NY, USA, 2010. Association for Computing Machinery.

[144] Mingxuan Yuan, Lei Chen, and Philip S. Yu. Personalized privacy protection in social networks. *Proc. VLDB Endow.*, 4(2):141–150, November 2010.

[145] Sean Chester, Bruce Kapron, Ramesh Ganesh, Gautam Srivastava, Alex Thomo, and Srinivasan Venkatesh. k-anonymization of social networks by vertex addition. In *ADBIS 2011, Research Communications, Proceedings II of the 15th East-European Conference on Advances in Databases and Information Systems, September 20-23, 2011, Vienna, Austria*, volume 789 of *CEUR Workshop Proceedings*, pages 107–116. CEUR-WS.org, 2011.

[146] Xiaowei Ying, Kai Pan, Xintao Wu, and Ling Guo. Comparisons of randomization and k-degree anonymization schemes for privacy preserving social network publishing. In *Proceedings of the 3rd Workshop on Social Network Mining and Analysis*, SNA-KDD '09, New York, NY, USA, 2009. Association for Computing Machinery.

[147] Lijie Zhang and Weining Zhang. Edge anonymity in social network graphs. *Proceedings of the 2009 International Conference on Computational Science and Engineering*, 4:1–8, 2009.

[148] Lian Liu, Jie Wang, Jinze Liu, and Jun Zhang. *Privacy Preservation in Social Networks with Sensitive Edge Weights*, pages 954–965. 2009.

[149] Sudipto Das, Ömer Egecioglu, and Amr El Abbadi. Anonymizing edge-weighted social network graphs. 2009. Available online: `https://sites.cs.ucsb.edu/~omer/DOWNLOADABLE/graph_anonymization_ICDE10.pdf`(accessed on 3 December 2020).

[150] Sudipto Das, Ömer Egecioglu, and Amr El Abbadi. Anonymizing weighted social network graphs. In *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)*, pages 904–907, 2010.

[151] Yidong Li and Hong Shen. On identity disclosure in weighted graphs. In *Proceedings of the 2010 International Conference on Parallel and Distributed Computing, Applications and Technologies*, PDCAT '10, pages 166—-174, USA, 2010. IEEE Computer Society.

[152] Shyue-Liang Wang, Zheng-Ze Tsai, Tzung-Pei Hong, and I-Hsien Ting. Anonymizing shortest paths on social network graphs. In *Proceedings of the Third International Conference on Intelligent Information and Database Systems - Volume Part I*, ACIIDS'11, page 129–136, Berlin, Heidelberg, 2011. Springer-Verlag.

[153] Donald E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. Association for Computing Machinery, New York, NY, USA, 1993.

[154] Chuan-Gang Liu, I-Hsien Liu, Wun-Sheng Yao, and Jung-Shian Li. K-anonymity against neighborhood attacks in weighted social networks. *Security and Communication Networks*, 8, 07 2015.

[155] Chris Clifton, Wei Jiang, Mummoorthy Murugesan, and M. Ercan Nergiz. Is privacy still an issue for data mining? In *Next Generation of Data Mining*, 2008.

Appendix A

Algorithms

Algorithm 8 Extended version of *kAnonymous* algorithm

Input: Undirected weighted graph G, parameter k, parameter m (optional)
Output: (k, m) anonymized graph G'

Read weighted graph data
if identifier exist **then**
    Replace them with random numbers/letters (Naive anonymization)
for each pair of nodes (u, v) (Initialization) **do**
    Store all neighbors weights (inexistent weights replaced by zero)
    Compute the distance between them by number of hops
while #nodes inito each $SN_i >= k$ (k-anonymity) **do**
    for each node $n_{1...w}$ **do**
        Select random node n_i
        Find the neighbors of n_i (1-hop)
        Find the neighbors' neighbors of n_i (2 hops)
        for all possible 2-hop neighbors **do**
            Select a random $2n_i$ 2-hop neighbor
            if nn_i have the same weight with n_i **then**
                Merge the two nodes in one supernode SN
                Create one superedge SE that connects the SN with the 1-hop neighbor
                Adjust the weight at the superedge
                Compute the information loss IL
                if the 1-hop neighbor has no other neighbors **then**
                    Merge the 1-hop neighbor to the SN

```
        else if they don't have the same weight then
            continue with another 2-hop neighbor
            Find the 2-hop neighbors that have the same weight
            Merge the two nodes in one supernode SN
            Create a superedge from the edges that connect the new supernode with all the other original nodes
            Adjust the weight at the superedge
        else if they don't have the same weight AND there is not any other 2-hop neighbor then
            Find the 2-hop neighbors that have similar weight
            Merge the two nodes in one supernode SN
            Create a superedge from the edges that connect the new supernode with all the other original nodes
            Adjust the interval of the possible weights of this superedge
        for all possible 1n_i 1-hop neighbors do
            if the 1n_i does not have another neighbor then
                Merge node 1n_i and the 2-hop neighbors into the same SN
                Compute the information loss
                Create a superedge from the edges that connect the new supernode with all the other original nodes
                Adjust the interval of the possible weights
    if n_i does not have a 2-hop neighbor or does not have a neighbor at all then
        if there exists a SN then
            Merge it into a SN that minimizes the information loss
        else
            Create a SN only with this node (other nodes will be added later)
for each supernode SN_{1...n} do
    Select random SN_i
```

if nodes into SN_i were connected in G **then**
Label the superedge with the probability P of edge existence
if $P = 1$ **then**
while $P < 0.8$ (threshold) **do**
Move n_k into another supernode that contains a neighbor in G, minimizing the information loss IL
Compute again the probability P of edge existence
for each pair of supernodes SN_j and SN_j **do**
if SN_i and SN_j contain nodes that were connected in G **then**
Label the superedge with the probability $P(SN_i -> SN_j)$ of edge existence
if $P(SN_i -> SN_j) = 1$ **then**
while $P(SN_i -> SN_j) < 0.8$ (threshold) **do**
Select SN_i and random node n_k
Move n_k into another supernode that contains a neighbor in G, minimizing the information loss IL
Compute again the probability $P(SN_i - . > SN_j)$ of edge existence
for each supernode $SN_{1...n}$ (Optional step) **do**
if there are not at least $m - 1$ supernodes that are connected with superdges that have the same probabilities and weights **then**
while ($\#SN_i$ with same PB and $WB >= m$) (m-anonymity) **do**
move nodes from SN_i to SN_j minimizing the information loss IL
Return G'
end

Appendix B

Tables

Table B.1: PPDM techniques comparison.

Article	Mining Model	Partition*	Environment	Privacy Method$^{\bullet}$	Attribute Type$^{\diamond}$	Execution†
Proposed Work	TAN	H and V	C2S, one miner, parties > 2	C	Nom and Num	I
[18]	Decision Trees	H	C2C, parties > 2	R	Num	E
[40]	EM clustering	H and V	C2C, parties > 2	C	Nd	T
[27]	Association Rules	H	C2C, parties > 2	C	Nd	I
[28]	Naive Bayes	H	C2C, parties > 2	C	Nom and Num	T
[19]	Decision trees	H	C2C, two parties	C	Nom	T
[68]	Decision trees	H	C2C, parties > 2	C	Nom	T
[63]	Association Rules	V	C2C, two parties	R	Bin	T
[29]	K2	V	C2C, two parties	C	Bin	T
[53]	Naive Bayes	H	C2S, one miner	C	Bin	I
[70]	Naive Bayes	H	C2S, two miners	C	Nd	T
[30]	Bayesian Nets	H	C2C	C	Nd	T
[54]	TAN	H	C2S, one miner	P	Num	I

* *DB partition: H=Horizontally, V=Vertically.*

$^{\bullet}$ *C=Cryptography, R=Randomization, P=Perturbation.*

$^{\diamond}$ *Nom=Nominal, Num=Numerical, Bin=Binary, Nd=Not defined.*

† *E=Empirical, T=Theoretical, I=Implemented.*

Table B.2: Comparison of privacy preserving techniques of graphs.

Article	Type of graphs	Anonymization Technique or Proposed Attack	Method	Problem (created / solved)	Attack (conducted / confronted)
Proposed Work	Weighted Undirected	Clustering-based	k-anonymity neighborhood grouping: nodes are groups in supernodes and edges into superedges based on the neighborhood similarities and edge weights.	Solved: Link disclosure Identity disclosure Edge weight disclosure	Confronted: Re-identification Link-based attacks Degree-based attacks Weight-based attacks Neighborhood-based attacks
Liu and Yang [124]	Weighted Undirected	Clustering – based	k-possible anonymity: groups nodes based on weight bag and the edge generalization by using weight intervals.	Solved: Identity disclosure	Confronted: Weighted based attacks Not referring
Li and Shen [96]	Weighted undirected	Perturbation – based	k-volume and k-histogram anonymity: modify the edge weights and node connections.	Solved: Identity disclosure	Confronted: Weight-based attacks (Volume and histogram attacks), Re-identification attacks
Li and Shen [151]	Weighted Undirected	Perturbation – based	Two algorithms presented to perturb the volume sequence.	Solved: Identity disclosure	Confronted: Volume attacks
Wang et al [152]	Weighted Undirected	Perturbation – based	k-anonymous path privacy: perturb minimal number of edge weights so that there are at least k indistinguishable shortest paths between the source and destination nodes.	Solved: Path anonymity Edge weight privacy	Confronted: Weighted based attacks Not referring
Liu et al [98]	Weighted (continuous weights) Directed	Perturbation – based	k-anonymous weight privacy, modify the edge weights, preserve shortest paths and shortest path lengths. The algorithm is based on random walk and matrix analysis to modify individual edge weights.	Solved: Edge weight disclosure	Confronted: Weighted based attacks Not referring
Liu et al [111, 148]	Weighted Undirected	Perturbation – based	Two perturbation strategies: Gaussian randomization multiplication and greedy perturbation algorithm. Perturb edge weights while preserving shortest paths and their lengths.	Solved: Edge weight disclosure	Confronted: Weighted based attacks Not referring

Table B.2: Comparison of privacy preserving techniques of graphs.

Article	Type of graphs	Anonymization Technique or Proposed Attack	Method	Problem (created / solved)	Attack (conducted / confronted)
Das et al [149]	Weighted Directed	Perturbation – based	Re-assign weights to edges so that the shortest paths of the original graph can be preserved. Edges are k-anonymous in their neighborhood.	Solved: Edge weight anonymization Identity disclosure	Confronted: Re-identification Weight-based attacks
Das et al [150]	Weighted Directed	Perturbation – based	Linear programming method to change edge weights while preserving shortest paths.	Solved: Edge weight anonymization Identity disclosure	Confronted: Re-identification Weight-based attacks
Backstrom et al. [3]	Unweighted Undirected	Passive and active attacks	An adversary learns whether edges exist or not between specific pair of nodes. Active : Create an distinguishable subgraph. Passive : try to find specific nodes in the released network, and discover the existence of edges among users to whom they are linked.	Created: Link disclosure	Conducted: Subgraph attacks
Bhagat et al. [142]	Unweighted Undirected Bipartite Labeled Nodes	Clustering-based Answer queries	"label list" approach: each node in the graph gets a list of possible identifiers, including its true identifier. "partitioning" approach: partitions the entities into classes, and describes the number of interactions at the level of classes, rather than nodes.	Solved: Link disclosure (edge safety condition)	Confronted: Link based attacks
Campan and Truta [106]	Unweighted Undirected Labeled Nodes (identifier, quasi-identifier, and sensitive attributes)	Clustering-based	Their method clusters the nodes based on attribute data and neighborhood and reveals only the number of edges within a group and between pairs of groups. The nodes have additional properties, which are generalized so that all nodes in the same cluster are indistinguishable in terms of their quasi-identifier attributes.	Solved: Content and identity disclosure	Not referring do not impose any restriction on the neighborhood attack graphs
Cheng et al [115]	Unweighted Undirected	Perturbation – based	k-isomorphism: form k pairwise isomorphic subgraphs by adding or deleting edges, partitioning the graph into k subgraphs with the same number of nodes.	Solved: Link disclosure Identity disclosure	Confronted: Re-identification attacks Subgraph attacks

Table B.2: Comparison of privacy preserving techniques of graphs.

Article	Type of graphs	Anonymization Technique or Proposed Attack	Method	Problem (created / solved)	Attack (conducted / confronted)
Chester et al [145]	Unweighted Undirected Labeled and unlabeled nodes	Perturbation – based	k-degree anonymity: modification techniques to the node set rather than the edge set, partitioning the degree sequence into subsequences of length at least k by adding dummy nodes.	Solved: Identity disclosure	Confronted: Degree based attacks
Clarkson et al [101]	Unweighted Undirected	Perturbation – based	k-degree anonymity: edge additions and deletions techniques.	Solved: Identity disclosure	Confronted: Re-identification Degree based attacks
Hay et al [109]	Unweighted Undirected	Perturbation – based	k-candidate anonymity: similarity of neighborhoods based on the candidate set of each node, performing a series of random edge deletions/additions such that the set of nodes is automorphically equivalent.	Solved: Identity disclosure	Confronted: Re-identification attacks Subgraph attacks
Hay et al [105]	Unweighted Undirected Unlabeled nodes	Clustering – based	Summarize graph topology in terms of node groups. Neighboring nodes are grouped in a supernode revealing only the number of edges among and within partitions and the number of nodes in each partition.	Solved: Identity disclosure Edge disclosure (adversaries knows the degree signatures)	Confronted: Re-identification attacks Structural attacks (do not impose any restriction on the neighborhood attacks)
Liu and Terzi [108]	Unweighted Undirected	Perturbation – based	k-degree anonymity: anonymize degree sequence, at least k- 1 other nodes in the graph with the same degree, edge additions/deletions.	Solved: Identity disclosure	Confronted: Re-identification (the adversary's background information consists only of node degrees)
Thompson and Yao [140]	Unweighted Undirected	Clustering – based Perturbation – based	k-anonymity based inter-cluster method, i-hop degree based approach, nodes are grouped in the same supernode based on a distance metric, each node within a supernode have the same degree.	Solved: Identity disclosure	Confronted: Re-identification Degree based attacks

Table B.2: Comparison of privacy preserving techniques of graphs.

Article	Type of graphs	Anonymization Technique or Proposed Attack	Method	Problem (created / solved)	Attack (conducted / confronted)
Tripathy and Panda [114]	Unweighted Undirected	Perturbation – based	K-anonymization of subgraphs, edge additions to establish isomorphism of neighborhoods. Two components with the same degree and having same adjacency matrices are isomorphic according to their structure.	Solved: Identity disclosure	Confronted: Neighborhood based attacks (even if the adversary has information not only about the immediate neighbors, but also about the nodes within finite number of hops from the target node.)
Wu et al [143]	Unweighted Undirected	Perturbation – based	k–symmetry anonymity: adding new edges and nodes, nodes are automorphically equivalent.	Solved: Identity disclosure	Confronted: Re-identification Structural knowledge (the adversary knows the entire graph, and the location of a node)
Ying and Wu [110]	Unweighted Undirected	Perturbation – based Randomization	Two randomization approaches that preserves the spectrum of the graph (1) randomly add one edge followed by deleting another edge and repeat this process for k times and (2) randomly switch a pair of existing edges.	Solved: Identity and subgraph disclosure	Confronted: Subgraph attacks
Zhang and Zhang [147]	Unweighted Undirected	Perturbation – based	Three heuristic algorithms that protect edge anonymity using edge swap or edge deletion. Degree-based edge swap, degree-based edge deletion and edge-based edge swap.	Solved: Link disclosure Edge anonymity	Confronted: Link identification attack
Zheleva and Getoor [100]	Unweighted Undirected Labeled edges	Clustering – based Perturbation – based	Edge deletion or addition and node-merging algorithms are used to ensure that nodes are indistinguishable in terms of their surrounding neighborhood.	Solved: Link disclosure	Confronted: Link re-identification attacks
Zhou and Pei [104]	Unweighted Undirected Labeled nodes (one attribute)	Perturbation – based Answer queries	Generalizing node labels and adding edges to create similar neighborhoods that are isomorphic, based on k-anonymity model.	Solved: Identity disclosure	Confronted: Neighborhood attacks (1-neighbor-graph attack)

Table B.2: Comparison of privacy preserving techniques of graphs.

Article	Type of graphs	Anonymization Technique or Proposed Attack	Method	Problem (created / solved)	Attack (conducted / confronted)
Zou et al [94]	Unweighted Undirected	Perturbation – based	k-automorphism: edge and node addition to create at least k similar isomorphic subgraphs.	Solved: Identity disclosure	Confronted: Subgraph attacks Structural attacks

Appendix C

Figures

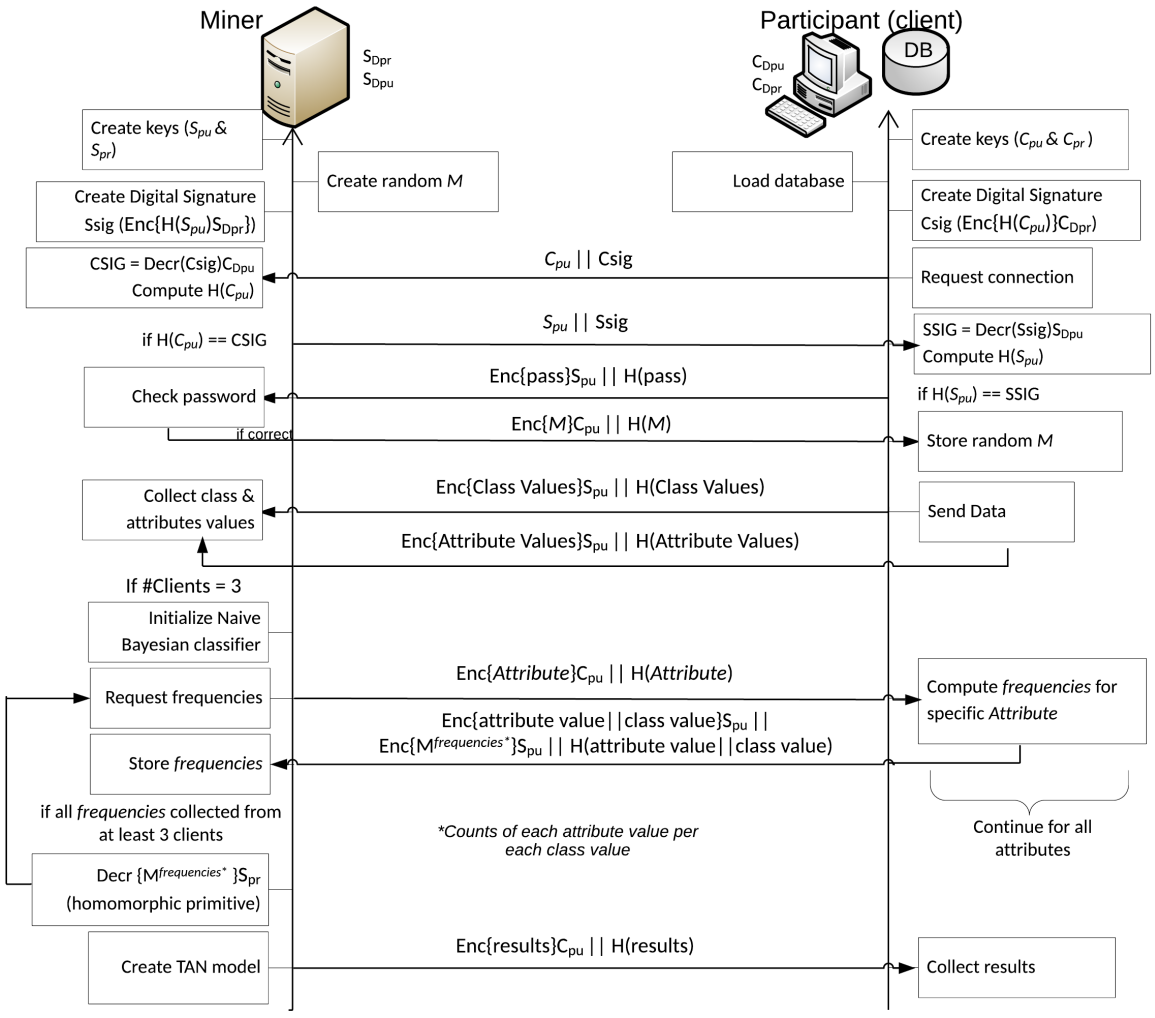


Figure C.1: Privacy preserving data mining protocol for distributed databases.

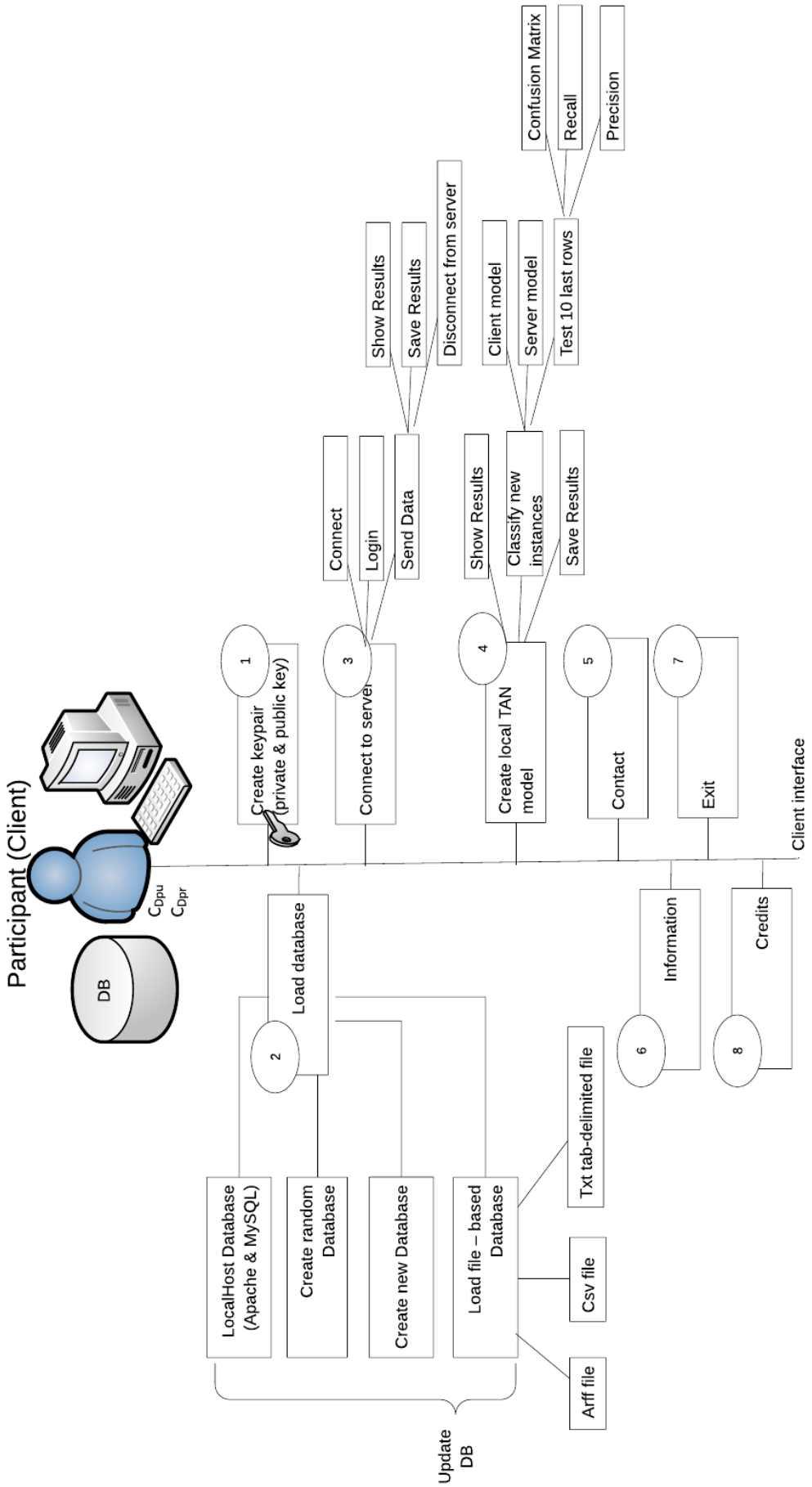


Figure C.2: Privacy preserving data mining protocol client interface.

Appendix D

Publications

Parts of the work described in this dissertation have been published in scientific journals and conference proceedings. A list of the related publications is the following:

1. Maria E. Skarkala, M. Maragoudakis, S. Gritzalis, L. Mitrou, Privacy Preserving Tree Augmented Naïve Bayesian Multi – party Implementation on Horizontally Partitioned Databases, TrustBus 2011 8th International Conference on Trust, Privacy and Security of Digital Business, S. Furnell, C. Lambrinoudakis, and G. Pernul, (eds), pp. 62 - 73, August 2011, Toulouse, France, Lecture Notes in Computer Science LNCS, Springer,

2. Maria E. Skarkala, Hannu Toivonen, Pirjo Moen, M. Maragoudakis, S. Gritzalis, L. Mitrou, Privacy Preservation by k-Anonymization of Weighted Social Networks, 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, P. Yu, K. Carley et al., (eds), pp. 423-428, August 2012, Istanbul, Turkey, IEEE CPS Conference Publishing Services, doi: 10.1109/ASONAM.2012.75.

3. Maria E. Skarkala, M. Maragoudakis, S. Gritzalis, L. Mitrou, PP-TAN: a Privacy Preserving Multi-party Tree Augmented Naive Bayes Classifier, SEEDA CECNSM 2020 5th South East Europe Design, Automation, Computer Engineering, Computer Networks and Social Media Conference, Corfu, Greece, 2020, pp. 1-8, doi: 10.1109/SEEDA-CECNSM49515.2020.9221844.

4. Skarkala, M.E.; Maragoudakis, M.; Gritzalis, S.; Mitrou, L. PPDM-TAN: A Privacy-Preserving Multi-Party Classifier. Computation 2021, 9, 6. https://doi.org/10.3390/computation9010006