Analysis of Topological Privacy Notions

Master's Thesis of

Frieder Haizmann

at the Department of Informatics
KASTEL – Institute of Information Security and Dependability
Practical IT Security

Reviewer: Prof. Dr. Thorsten Strufe
Second reviewer: Prof. Dr. Jörn Müller-Quade
Advisor: M. Sc. Patricia Guerra-Balboa

15. April 2024 – 15. October 2024

Karlsruher Institut für Technologie
Fakultät für Informatik
Postfach 6980
76128 Karlsruhe

Abstract

The current state-of-the-art notions in statistical disclosure control face some problems in real-world data privacy scenarios, by either making best-case assumptions or potentially degrading the utility of the data. However, in 2017 Erdmann introduces two new syntactic privacy notions for Relations: *Attribute Privacy* and *Association Privacy*. These notions are defined over the topology of the released data. Notably, they do not include best-case assumptions, and typically syntactic notions provide good data utility. It is thus worthwhile to analyze their benefits and drawbacks in the context of existing notions. Nevertheless, up to this point, no formal analysis has been performed, and no mechanism has been given for these notions.

Therefore, in this work, we formally analyze the topological notions, providing theoretical proofs about their protections, and compare them to the current state-of-the-art privacy notions in statistical disclosure control. Further, we provide a mechanism to achieve Attribute Privacy, whose utility we empirically analyze. Overall, we reveal multiple flaws inherent to the topological notions, and come to the conclusion that they are not suited as a generalized replacement for the existing relational data state-of-the-art. Additionally, as Association Privacy has the protection goal of hiding associations between people, we provide an analysis of Association Privacy in the context of graph data privacy. We find that the existing syntactic graph privacy notion of k-security provides stronger protections.

Kurzfassung

Der etablierte Stand der Technik im Feld der statistischen Offenlegungskontrolle stößt in realen Datenschutzszenarien auf Probleme, da die etablierten Datenschutzbegriffe entweder auf Best-Case-Annahmen beruhen oder die Nutzbarkeit der Daten potenziell beeinträchtigt wird. In 2017 führt Erdmann zwei neue syntaktische Datenschutzbegriffe für Relationen ein: *Attribute Privacy* und *Association Privacy*. Diese Begriffe werden über die Topologie der freigegebenen Daten definiert. Sie treffen keine Best-Case-Annahmen. Des Weiteren bieten in der Regel syntaktische Datenschutzbegriffe Daten mit guter Nutzbarkeit. Daher lohnt es sich, die Vor- und Nachteile dieser topologischen Begriffe zu analysieren, und mit den bestehenden Begriffen zu vergleichen. Allerdings wurde bisher weder solch eine formale Analyse durchgeführt, noch wurde ein Mechanismus für diese Begriffe entwickelt.

Daher analysieren wir in dieser Arbeit jene topologischen Datenschutzbegriffe formal. Wir legen theoretische Beweise für ihre Schutzgarantien dar, welche wir mit denen des etablierten Stands der Technik vergleichen. Darüber hinaus entwickeln wir einen Mechanismus, welcher Attribute Privacy auf einen gegebenen Datensatz sicherstellt, und analysieren dessen Nutzen empirisch. Insgesamt zeigen wir mehrere Schwachpunkte der topologischen Datenschutzbegriffe auf, und kommen zum Schluss, dass diese Begriffe kein geeigneter Ersatz für die etablierten Begriffe darstellen. Das Schutzziel des Begriffs Association Privacy, ist es Verbindungen (Assoziationen) zwischen Personen zu verbergen. Da dies Schutzzielen von Datenschutzbegriffen für Graphdaten entspricht, bieten wir zusätzlich eine Analyse von Association Privacy im Kontext des Datenschutzes von Graphdaten. Dabei zeigen wir, dass der existierende syntaktische Datenschutzbegriff für Graphdaten k-Security stärkere Schutzgarantien bietet.

Contents

List of Figures

List of Tables

1. Introduction

Data collection, aggregation, and publication is a massive field with various sources and applications: One such application is forecasting competitions, in which companies release datasets, and challenge the data mining community to achieve a better prediction algorithm or machine learning model on that dataset. Historically, one such challenge was the Netflix prize dataset [9], in which Netflix challenged data scientists to beat the accuracy of their recommendation system, providing a large movie rating dataset. More recently, the online data science platform Kaggle has become a popular platform among researchers [10] for such challenges. These challenges include, for example, grocery store chain sales forecasting, or predicting web-traffic [10]. Kaggle claims on their homepage [34] 385 thousand datasets and over 20 million users. Another public and highly studied dataset is the U.S. census [3]. Besides relational databases, which typically represent a record of attributes for a number of individuals, graph data, which is able to represent a network of individuals, also has a wide range of uses. These range from social network analysis, to financial services, supply chains, and healthcare [60]. The latter includes, for example, graph modelling for tracking the COVID-19 pandemic spread [2].

However, data that is insufficiently anonymized bears risks of disclosing sensitive information about individuals whose data has been used, especially when used in conjunction with background data. As an example, the aforementioned Netflix prize dataset was deanonymized by linking records with the Internet Movie Database [72]. This enabled the researchers to identify records of known users in the Netflix dataset, and uncover potentially sensitive information such as apparent political beliefs [72]. Another such linkage was performed by [79], who linked health records collected and distributed by an insurance commission with voter registration data, by using the overlapping attributes of zip-code, birthdate, and sex.

In order to avoid such attacks and more broadly protect a participant's privacy when it comes to data analysis, the field of statistical disclosure control (SDC) has emerged. The goal of SDC is to be able to disclose statistical data in such a way, that privacy of the participants can still be ensured [90]. To do so, we need to measure the privacy of the SDC process. Consequently, different privacy notions have been introduced. These are grouped in syntactic notions, which impose conditions over the published data, and semantic notions, which apply to release mechanisms that answer data queries. Of those, in the context of relational data, k-Anonymity[79], and its extensions ℓ-Diversity [65] and t-Closeness [57] have established themselves as the state-of-the-art (SOTA) for syntactic notions, and ε-Differential Privacy (DP) has established itself as the SOTA semantic notion. Subsequently, these notions have been formally analyzed thoroughly. In the context of graph data, there seems to not be an agreed upon state-of-the-art yet.

Despite being state-of-the-art, k-Anonymity (and its extensions) exhibit a major flaw when it comes to privacy guarantees: These notions rely on the strong assumption, that a data publisher correctly identifies the structure of an attacker's background knowledge. If this assumption about the attacker's background knowledge is broken, the privacy guarantees of these notions no longer hold. On the other hand, DP makes no such

assumptions and offers strong privacy guarantees. However, DP offers significantly worse data utility, than the syntactic SoTA. This means there is a trade-off to be had, between strong privacy guarantees, and data utility. This also has been codified by [51] into the "no-free-lunch theorem", which states that it is impossible to provide privacy and utility without making assumptions about how the data are generated.

As a solution to previous notions' problems, a set of two new privacy notions for relations is proposed by Erdmann in 2017 [33]. These are the notions Attribute Privacy and Association Privacy, which leverage the topology of the underlying data. The two topological notions are introduced in duality to each other, however their protection goals differ: Attribute Privacy aims to prevent an inference attack, where an attacker with background knowledge about some attributes of their target can infer more attributes. Association Privacy is intended to protect the relationship between people, which is more in line with protection goals we would expect from notions for graph data. These new notions could potentially offer a better trade-off proposition: They might offer stronger privacy guarantees than k-Anonymity, as they do not share its assumption about the attacker's background knowledge. At the same time, they might offer better utility than DP, as they are syntactic notions, which typically exhibit better utility than semantic notions. Further, a later paper [87] claims that a combination of Attribute Privacy and DP can rectify issues DP has with correlated data.

These notions have stayed relatively unexplored, and in particular, neither a formal analysis of them nor a comparison with respect to the previous ones has been given.

Consequently, this thesis aims to understand whether the topological notions presented by [33] offer a favorable trade-off between privacy and utility over the state-of-the-art, and to find other potential practical applications of these notions. We do this by answering subsequent research questions. The first of which is as follows: How does the privacy provided by the topological notions compare to the privacy provided by the current state-of-the-art SDC notions for relational data? Because notions have different (implicit) assumptions, we begin answering this by comparing and contrasting the assumptions with each other, judging the impact on privacy. We further pose follow-up research questions to compare the privacy provided by the topological notions with the state-of-the-art. Additionally, we answer whether the topological notions stand in any direct relationship with the SoTA. We do so by introducing a systematic way to convert an arbitrary database into a relation, and then we either prove a non-implication between notions via counterexamples or prove an implication.

Furthermore, we explore the resilience of topological notions with respect to the most common attacks on private data, recompiled in [37], by conducting a theoretical analysis of attack resilience. For the negative results, i.e., attacks that topological notions fail to protect against, we perform an example of the target attacks on topologically protected data, showing directly the success despite the notion. Finally, we compare our resilience results for the topological notions with existing results for the SoTA.

We are also interested in whether attribute privacy could be utilized in use-cases such as continuous data publishing, or sequential releases. For these use-cases, we need to understand the composition properties of Attribute Privacy. There are known composition results for ε-DP, and negative results for k-Anonymity. We investigate the ways in which Attribute Privacy composes. We do this by introducing new theorems, about a set of conditions on the relation, that are equivalent to Attribute Privacy. Moreover, we examine whether these conditions are still met under parallel or sequential composition, and give a counterexample if the result is negative.

We also want to investigate the claims of [87], and analyze whether a combination of Attribute Privacy and DP can solve the problem with correlated input data in DP. For this question, we analyze the effect of input-correlated data on Attribute Privacy.

We notice that the stated goal of Association Privacy is surprisingly similar to a protection goal we would expect in the context of graph data. For this reason, we want to explore, if Association Privacy has a place amongst graph data SDC notions. To investigate this answer, we first need to establish what the SOTA in graph data notions is. This helps us identify common attack-models and graph-privacy notions with similar protection goals as Association Privacy. We introduce a method to create a graph from a relation protected by Association Privacy. We analyze if such graphs are then vulnerable against the attack models on graph data previously found, by performing such attacks on those graphs. With this knowledge, we can then judge if there are existing syntactic notions that provide better graph-data privacy than Association Privacy. We also investigate, if Association Privacy can be used to protect data that are initially given as graph data, by mapping it into a relation, protecting that relation, and mapping it back into a graph.

Finally, to ascertain the privacy-utility trade-off, we need to analyze the utility achievable under the topological notions. As utility is not a property of a notion, but a property of a mechanism achieving a notion, and for lack of existing mechanisms for the topological notions, we develop a mechanism for Attribute Privacy. We base this mechanism on the principle of modifying the topological structure corresponding to the relation. With this mechanism, we perform utility experiments whose result we compare with the corresponding utility results of ε-Differential Privacy.

We list our contributions:

- In the context of relational data:
  - We classify and formalize the privacy provided by the topological notions, with respect to the SOTA, categorizing their attack resilience against most known attacks. With this we show that Association Privacy is not suited to protect relational data, and that Attribute Privacy has better assumptions than the syntactic SOTA but only offers privacy on a narrow attack vector.
  - We provide novel composition theorems for Attribute Privacy and discuss the usability of Attribute Privacy in the context of continual observation. We find that Attribute Privacy holds under parallel composition, but breaks under sequential composition. Further, we find Attribute Privacy is not suited for data under continual observation.
  - We analyze the influence correlated input data can have on the protections given by Attribute Privacy, and show that depending on the type of correlation Attribute Privacy falters. We consequently judge that Attribute Privacy is likely not suited to supplement DP.
  - We develop a mechanism which achieves Attribute Privacy. We analyze the utility it provides in comparison with the Laplace-Mechanism for DP. In the analysis we show that the Attribute Privacy mechanism performs significantly worse than Laplace, and we provide arguments why similar mechanisms would have comparable results.

- In the context of graph data:
  - We give an overview of SOTA syntactic notions, grouped by their protection goals and assumed background knowledge.

- We introduce a method to create a graph from an association-private relation, and perform an attack analysis on such graphs with graph data adversaries. We show that the existing syntactic notion of *k*-security[18] provides better graph-data privacy, than Association Privacy.

- We further show the infeasibility to directly apply Association Privacy to data, initially given as a graph.

In the following, we outline the structure of this thesis. We first provide background information in chapter 2. There, we introduce and explain the current state-of-the-art of notions in SDC, and elaborate on their challenges in real-world data privacy scenarios. Additionally, we provide topological foundations that are necessary for the definition and analysis of the topological notions. We introduce the topological notions that are discussed in the remainder of the thesis, and lay out their explicit and implicit assumptions. Next, in chapter 3, we present additional definitions, lemma and theorem, we developed, which we use for later proofs. Then, in chapter 4 we perform our analysis of the topological notions in the domain of relational data. There, we perform our composability analysis. We compare the topological notions with the state-of-the-art notions. This includes a comparison of the assumptions these notions hold, an analysis of the relationships between notions, and an attack resilience analysis, of SoTA attacker against the topological notions. Further, we analyze if combining Attribute Privacy with ε-Differential Privacy can solve DP's problem with correlated input data. In chapter 5 we answer our research questions regarding graph data. We summarize the existing protection goals for graph data and assumed background knowledge. Then we give a brief summary of existing SoTA graph privacy notions, their protection goals, and assumed background knowledge. Next, we formalize a way to create a graph from an association private relation. Further, we formalize the adversary for the graph privacy protection goal closest to Association Privacy's protection goal. We perform attacks with this adversary against graphs generated from association private relations. We contrast this with an existing graph privacy notion. In chapter 6, we introduce our mechanism to achieve Attribute Privacy. Subsequently, in chapter 7 we analyze its utility and compare it to the utility provided by ε-DP. Finally, we close the thesis in chapter 8, in which we provide the conclusions we draw from our work.

We provide additional information on existing informal analysis of graph data algorithms for privacy in Appendix A.

Our code is available in the following repository `https://gitlab.kit.edu/urpyg/thesis-topology-of-privacy/`, in the `Scripts` subfolder. Images of simplicial complexes are generated using *GeoGebra* (`www.geogebra.org`) and *PolyMake* [39].

1.1. Related Work

We were not able to find usages of the attribute and Association Privacy notions in other papers with one exception. A flaw of ε-Differential Privacy is pointed out by [87]: Correlated attributes can allow a successful attribute linkage attack on data that is protected by DP. The authors suggest a combination of a relaxation of Attribute Privacy and ε-Differential Privacy to solve this issue. They provide a mechanism and utility analysis, however the exact protections given are not formally analyzed. We further investigated this paper's claims.

A systematization of privacy notions for relational data is presented in [37]. While this work does not discuss the topological privacy notions, it does analyze and compare ε-Differential Privacy, k-Anonymity and its extensions, as well as other privacy notions (which will not be discussed in the thesis). It also provides attack models, which we use to test the topological notions against.

There are multiple surveys concerned with graph data privacy: [93, 1] iterate over several mechanisms, however the corresponding notions are not analyzed or compared with each other. A more recent survey [60] is focussed on "provable" (semantic) graph privacy notions. The authors thus compare some differential privacy adaptions for graph data and pufferfish privacy. They dismiss syntactic notions as not provable but refer to [53] for a discussion of syntactic notions. However [53] is also only concerned with "techniques", and devotes only a paragraph of a short description to each of the 35 discussed techniques. While this means these surveys lack a formal comparison of syntactic graph privacy notions, surveys [53, 1] and paper [62] do provide an informal classification of privacy breaches in graph data, which appears consistent between the works. Further [93] lists different types of background knowledge, that are assumed by various algorithms for private graph data.

2. Background

In this chapter, we introduce concepts used throughout the thesis. This includes the state-of-the-art SDC notions on relational data, topological concepts needed to define the topological notions and later proofs, the topological notions themselves, as well as properties of these notions. In Table 2.1 we list notations we use throughout the thesis.

Notation	Description
X	The set of all individuals in a given relation
Y	The set of all attributes in a given relation
R	A relation on individuals and attributes $X \times Y$
x	Usually an individual
y	Usually a (binary) attribute
X_y	The set of all individuals that share the attribute y in a given relation
Y_x	The set of attributes, the individual x has in a given relation
Φ_R	The attribute complex of relation R
Ψ_R	The association complex of relation R
σ	A set of individuals; simplex representing individuals in Ψ_R
γ	A set of attributes; simplex representing attributes in Φ_R
ϕ_R	Operator mapping sets of individuals to shared attributes
ψ_R	Operator mapping sets of attributes to sharing individuals
$\mathcal{M}$	A mechanism ensuring some privacy notion

Table 2.1.: Notation used in this thesis

2.1. State-of-the-Art SDC Notions for Relational Data

SDC privacy notions are classified in two categories: syntactic and semantic. Syntactic notions impose conditions over published databases, while semantic notions are defined over mechanisms that release the data. In this latter case, a database is considered protected when it corresponds to the output of a private mechanism. Orthogonal to this, the data to be protected can have different types. *Relational data* can be represented using tables, where rows represent individuals, and columns represent attributes. *Graph data* can be represented using a graph, usually with vertices representing individuals. In the thesis, we mainly focus on relational data, however we devote chapter 5 to graph data.

For relational data, the current state-of-the-art of syntactic notions are k-Anonymity [79] and its extensions ℓ-Diversity [65] and t-Closeness [57]. The semantic state-of-the-art notion is ε-Differential Privacy [25, 26]. In this section, we explain the intuition and give formal definitions for these notions.

For most of the thesis, we discuss data given as Individual-Attribute-Relations, which we define in Definition 2.4.1. In these relations an individual can either have an attribute or

not have it. For example, individual Bob can have the attribute "high income", describing that fact about him. However, the SOTA notions are defined over more general relational databases, in which Bob's value for the attribute "income" might be "60000". For this reason, we define our notation over databases here.

In this thesis, we assume that a database D consists of a single table, with rows corresponding to individuals of an individual set $X = \{x_1, \ldots, x_n\}$, and columns representing attributes $y_1, \ldots, y_m$. We note $v \coloneqq D[x_i, y_i]$ the value of attribute y_i for individual x_i. In our above example, x_i is Bob, y_i is "income", and $v = 60000$.

The row representing an individual, which contains this individual's values for the attributes, is called a record. Further, given two databases D_1 and D_2 that have the same set of attributes, we use set notation to signify relationships between sets of records contained in those databases. So, for example, we write $D_1 \subseteq D_2$ to mean all records in D_1 are also contained in D_2.

The idea behind k-Anonymity is that a certain combination of attributes is enough to identify an individual within a database, which allows an attacker to then read the target's sensitive attributes. This combination of attributes is known as a quasi-identifier (QID), and although it is not an explicit identifier such as a name or address, it can be used to identify an individual.

k-Anonymity assumes for a given database table T, that T is a subset of some larger population Ω, such that each record $t_i \in T$ represents an individual from Ω [65].

Definition 2.1.1 (Quasi-Identifier [65])**.** A set of nonsensitive attributes $\{q_1, \ldots, q_w\}$ of a table is called a *quasi-identifier*, if these attributes can be linked with external data, to uniquely identify at least one individual in the general population Ω.

For example, the set of attributes $\{\text{zip-code}, \text{birth date}, \text{sex}\}$ has been used by Sweeney to identify the governor of Massachusetts in a dataset of medical records, using an external dataset of voter registration records [79, Example 1]. k-Anonymity assumes that this set of attributes, which identifies an individual, QID, is known to the analyst.

Definition 2.1.2 (k-Anonymity [65])**.** A database D for individuals X and attributes Y satisfies k-Anonymity, if for every record $t \in D$, there exist $k - 1$ other records $t_1, t_2, \ldots, t_{k-1} \in D$, such that

$$D[t, y] = D[t_1, y] = D[t_2, y] = \ldots D[t_{k-1}, y], \quad \text{for all } y \in \text{QID} .$$

So, the goal of k-Anonymity is to create anonymity groups, such that every QID that appears in the database appears at least k times. This way, knowing a target's QID will not enable an attacker to uniquely identify the target anymore. In a dataset D, which has been anonymized with k-Anonymity, an *equivalence class* is a set of records that have the same values for the quasi-identifier [57].

However, if within an equivalence class the values of a sensitive attribute S are the same for all individuals in that equivalence class, the privacy of that sensitive attribute is not given anymore. In such a case, an attacker is able to read the sensitive attributes of their target, despite not being able to tell, which individual within the group the target is.

To prevent this kind of attacks, the additional notion of ℓ-Diversity was introduced. ℓ-Diversity requires each anonymity group to have sensitive attributes, which are diverse by a metric parameterized over ℓ, for example, entropy $\leq \log \ell$.

Definition 2.1.3 (ℓ-Diversity Principle, based on [65, 57])**.** Given a k-Anonymity table D, whose QID attributes generalize to q^*. An equivalence class is ℓ-diverse if it contains

at least ℓ well-represented values for the sensitive attribute S. A database is ℓ-diverse if every equivalence class of the table is ℓ-diverse.

Definition 2.1.4 (Entropy ℓ-Diversity [57])**.** The entropy of an equivalence class E is defined as

$$\text{Entropy}(E) = -\sum_{s \in S} p(E, s) \log p(E, s) \,.$$

Where S is the domain of the sensitive attribute, and $p(E, s)$ the fraction of records in E, that have sensitive value s. A table has entropy ℓ-diversity, if for every equivalence class E $\text{Entropy}(E) \geq \log \ell \,.$

It can still happen that a sensitive attribute is over- or underrepresented within a group when compared to the whole database. This is rectified by t-Closeness, which ensures the statistical distribution of sensitive attributes within a group resembles the distribution of the whole dataset.

Definition 2.1.5 (t-Closeness Principle [57])**.** An equivalence class has t-Closeness, if the distance between the distribution of a sensitive attribute in this class, and the distribution of the attribute in the whole table is no more than a threshold t. A table has t-Closeness, if all equivalence classes have t-closeness.

The state-of-the-art for semantic notions is ε-Differential Privacy (DP). The goal of ε-Differential Privacy is to ensure for any participant, that their (non)-presence in a database is concealed. Intuitively, if an individual removed their record from the dataset, no output of the mechanism should meaningfully change. More precisely, it ensures that for any two neighbor databases (databases that differ from each other in one entry, which represents an individual) the output is ε-indistinguishable. ε-indistinguishably limits, how much two random variables can differ from each other for any possible event.

Definition 2.1.6 (ε-Differential Privacy [25])**.** A randomized function $\mathcal{K}$ gives ε-Differential Privacy, if for all datasets D_1 and D_2 differing on at most one record, and all $S \subseteq \text{Range}(\mathcal{M})$,

$$\Pr[\mathcal{K}(D_1) \in S] \leq \exp(\varepsilon) \cdot \Pr[\mathcal{K}(D_2) \in S] \,.$$

ε is also called the privacy budget. Larger ε values correspond to a higher chance of a correct inference by the attacker, and accordingly also correspond to less privacy. However, in most DP mechanisms, a larger privacy budget allows for better utility results.

The aforementioned notions encounter certain challenges when applied to real-world data privacy scenarios. Syntactic notions can be achieved while preserving high data utility. However, the privacy offered is a best-case scenario, in which an attacker with more background or different QID knowledge can obtain the target information even if these notions are satisfied. ε-Differential Privacy presents a worse-case metric, which is much more robust in terms of privacy, but the trade-off between privacy and utility is not always easy. The majority of DP mechanisms work for query protection, and the utility loss increases dramatically when we try to adapt them for microdata publication.

2.2. Composition

Composability in SDC is the property that two or more databases or mechanisms, which individually satisfy a privacy definition still satisfy the definition collectively, typically with a change in parameters [21].

This becomes relevant in the semantic case when multiple queries need to be answered over the same database. Such queries could, for example, be "how many patients in zip-code z have a viral infect" for multiple zip-codes z.

However, composition is also relevant in the context of syntactic notions. For example, for a medical study over a set of individuals, a dataset with k-Anonymity is published, and a year later, for a follow-up study over the same set of individuals, a new dataset that also satisfies k-Anonymity is published. It would be desirable, however not trivial to ensure that the combined dataset still ensures k-Anonymity. Note that while many composability theorems are generalized to an arbitrary number of data releases, for simplicity, in this thesis we mostly focus on the case in which two data releases compose.

The most well-known composability theorems are parallel and sequential composition [69]. *Parallel composition* analyzes a sequence of data releases, where the data used for every release comes from disjoint (sub-)databases. More precisely, for every release the sets of input records (individuals) are disjoint. Typically (as in most use cases) all releases contain information about the same set of attributes. The full original database D can be seen as being split horizontally into databases A and B, as pictured in Figure 2.1.

Figure 2.1.: Structure of a database for parallel composition

We adapt the definition of a partitioning function from [42, 58]: A *k-partitioning function* $p = \{p_1, \dots, p_k\}$ is a function, which maps a database to a database of the same domain, such that for a given database D, $p_i(D) \subseteq D$ and $p_i(D) \cap p_j(D) = \emptyset$, for $i \neq j$. Then we differentiate in the semantic and syntactic scenario:

- *Semantic*: the two corresponding releases would be $\mathcal{M}_A \circ p_1(D) = \mathcal{M}_1(A) = A'$ and $\mathcal{M}_A \circ p_1(D) = \mathcal{M}_2(B) = B'$. Given that $\mathcal{M}_1, \mathcal{M}_2$ satisfy the privacy notions Γ_1, Γ_2, the question is, which privacy notion is satisfied by $\mathcal{M} := (\mathcal{M}_1 \circ p_1, \mathcal{M}_2 \circ p_2)$.

- *Syntactic:* the two corresponding releases would be A' and B' satisfying the privacy notions Γ_1, Γ_2, and the question is which privacy notion is satisfied by the joint database $D' = A' \cup B'$.

In the semantic setting the mechanisms would fulfil the notion, and we would analyze for example the bounds of ε both releases give.

Theorem 2.2.1 (Parallel Composition for Differential Privacy, adapted from [69])**.** *Let D be a database and $p = \{p_1, \dots, p_k\}$ a k-partitioning function on D. Let $\mathcal{M}_1, \dots, \mathcal{M}_k$ be mechanisms, where $\mathcal{M}_i$ is ε_i-Differentially Private.*

The mechanism

$$\mathcal{M} := (\mathcal{M}_1 \circ p_1, \dots, \mathcal{M}_k \circ p_k)$$

is ε-Differentially Private with $\varepsilon = \max_{i \in 1..k} \varepsilon_i$

The intuition behind this is that the presence or absence of a given individual in D, will at most be represented in one of the sub-databases $p_i(D_i)$, which means it at most influences the output of $\mathcal{M}_i$. In the worst case, the $\mathcal{M}_i$ whose output is influenced is ε_i-DP with the biggest ε_i.

Sequential composition analyzes a sequence of data releases, where the data used for each entry in the sequence of data releases comes from potentially overlapping sub-databases. Here, the releases typically contain data of the same set of records, and potentially intersecting sets of attributes. The full original database D can be seen to be split vertically into databases A and B, as pictured in Figure 2.2. Two cases for composing

Figure 2.2.: Structure of a database for sequential composition

databases are

- $\mathcal{M}(A) = A'$ and $\mathcal{M}(B) = B'$: The first release is based on the left part of the database, the second release is based on the right part.

- $\mathcal{M}(A) = A'$ and $\mathcal{M}(D) = D'$: The first release is based on the left part of the database, the second release is based on both parts. Because A and B are allowed to overlap in attributes, this is a special case of the former.

We again differentiate between the semantic and syntactic scenario:

- *Semantic*: given that $\mathcal{M}_1, \mathcal{M}_2$ satisfy the privacy notions Γ_1, Γ_2, the question is, which privacy notion is satisfied by $\mathcal{M} := (\mathcal{M}_1 \circ f, \mathcal{M}_2 \circ g)$, where f and g are functions where f is defined such that $f(D) = A$, and g is defined such that $g(D) = B$.

- *Syntactic:* the two corresponding releases would be A' and B' satisfying the privacy notions Γ_1, Γ_2, and the question is which privacy notion is satisfied by the joint database D', in which each record $r' \in D'$ is the product of combining both corresponding records $r_1' \in A'$ and $r_2' \in B'$.

Additionally, for sequential composition of relations, we will later discern between *static releases* and *dynamic releases* (Definition 4.1.1). This is for the case where each release contains the superset of attributes of the previous release. In static releases, already released data is static and cannot be changed. Only new data (for new attributes) is added. In contrast, dynamic sequential releases allow a later release to contain information that conflicts with previously released data. Note, that the source data does not change, but a release mechanism likely changes some attributes, and inconsistencies between releases might allow an attacker to infer the original state of the data.

ε-Differential Privacy composes sequentially. A composition theorem is given for the case, in which the source of both releases is the full database.

Theorem 2.2.2 (Sequential Composition for Differential Privacy [27]). *Let $\mathcal{M}_1$ be a ε_1-Differentially Private mechanism, and $\mathcal{M}_2$ be a ε_2-Differentially Private mechanism, their combination*

$$\mathcal{M} := (\mathcal{M}_1, \mathcal{M}_2)$$

is $\varepsilon_1 + \varepsilon_2$-Differentially Private.

The intuition behind this theorem is that the presence or absence of an individual in the source data affects the output of both mechanisms. However, the individual's probabilistic influence on the output of $\mathcal{M}_i$ is bounded by $\exp(\varepsilon_i)$ ($i \in \{1, 2\}$), which means by taking the likelihood of both events (multiplying probabilities), the probabilistic influence on the combined output is bounded by $\exp(\varepsilon_1) \cdot \exp(\varepsilon_2) = \exp(\varepsilon_1 + \varepsilon_2)$. This also covers cases in which the input data to the composing mechanisms is not the full database, as a ε_i-DP mechanism $\mathcal{M}'_i = \mathcal{M}_i \circ f_i$, that discards information of the input database is trivially constructed from $\mathcal{M}_i$.

There are multiple results that show that *k*-Anonymity does not generally compose [78, 38, 77], these attacks rely on inconsistent choice of anonymity classes between releases. This means an individual can appear in different anonymity classes, and an attacker can, for example, compare sensitive attributes that are common among all of those anonymity classes. Some of these attacks are clearly showing a breakdown of privacy guarantees under sequential composition, as they rely on the entire database D (or overlapping sub-databases) to be released multiple times [78, Example 3].

All remaining *k*-Anonymity composition attacks we found rely on the condition, that at least one individual is present in multiple independently recorded databases that are released [38, 77]. For example, the same patient appearing in two Hospitals' databases. This is arguably parallel composition, as given two independently recorded databases D_1 and D_2, each containing a record representing the data of the same individual, $r_1 \in D_1, r_2 \in D_2$, it is not necessarily the case that $r_1 = r_2$. The record r_2 could, for example, contain additional attributes that are not relevant to the attack, like a secondary sensitive attribute, that is not captured by r_1. With an alternate definition of database overlap, where $D_1 \cap D_2 \neq \emptyset$ if both databases capture data about the same individual, the releases on which these attacks work could be classified under sequential composition.

We would like to acknowledge related concepts, which we will not be using: Specifically for syntactic notions [75], introduces the dichotomy between *continuous data publishing* (based on [13]) and *sequential release publishing* (based on [88]). In continuous data publishing the set of attributes is fixed, and records (rows; individuals) are modified, while in sequential release publishing the set of individuals is fixed and columns (attributes) are modified. In this context, modification can mean insertions, deletions, reinsertions, and updates.

For semantic notions [42] define the terms independent and adaptive sequential releases, which stand orthogonal to the parallel and sequential composition concepts. There the more general term, adaptive composition, means that any mechanism of the sequence also receives the outputs of the preceding mechanisms as input. And the less general term, independent composition, means that the outputs of the mechanisms are independent of each other.

2.3. Topological Concepts

In this section we explain topological concepts, needed to define the topological privacy notions, and through the thesis.

Definition 2.3.1 (Independent set of points [67, Definition 2.3.1]). A set of $n + 1$ points $a^0, a^1, \ldots, a^n$ in R^m is said to be independent if the vectors $a^1 - a^0, a^2 - a^0, \ldots, a^n - a^0$ are linearly independent.

Definition 2.3.2 (n-simplex [67, Definition 2.3.2]). A n-Simplex σ is the set of points

$$\sum_{i=0}^{n} \lambda_i a^i \,,$$

where $a^0, a^1, \ldots, a^n$ are independent in some Euclidean space R^m, and the λ_i are real numbers, such that $\lambda_i \geq 0$ for all i and $\sum_{i=0}^{n} \lambda_i = 1$. The points $a^0, a^1, \ldots, a^n$ are called *vertices* of σ and are said to *span σ*.

If $a^{i_0}, a^{i_1}, \ldots a^{i_r}$ is any subset of vertices of σ, the subspace of σ of those points linearly dependent on $a^{i_0}, a^{i_1}, \ldots a^{i_r}$ is called a *face* of σ. A face could be empty or the whole of σ. It is called *proper* if it is neither of these. Finally, n is called the dimension of σ.

In Figure 2.3 we find one way of simplex representation by drawing each simplex as their equivalent shape. E.g. a simplex consisting of two points becomes a line between these two points, a simplex consisting of three a triangle. Throughout the thesis, we will

(a) σ_0 (b) σ_1 (c) σ_2 (d) σ_3

Figure 2.3.: Examples of n-simplexes with $n = 0, \ldots, 3$.

denote an n-simplex σ by the set of $n + 1$ vertices, that span it. E.g. $\sigma := \{a, b, c\}$ is the 2-simplex, spanned by the 3 independent vertices with the labels a, b, and c respectively. This notation also makes it easier to describe a face of a simplex: Given two sets of vertices $\boldsymbol{\mu}$ and γ, each spanning a simplex, then the simplex spanned by $\boldsymbol{\mu}$ is a face of γ, iff $\boldsymbol{\mu} \subseteq \gamma$. Referring to the simplices by their vertex sets, $\boldsymbol{\mu}$ is a proper face of γ, iff $\boldsymbol{\mu} \neq \emptyset$, and $\boldsymbol{\mu} \subsetneq \gamma$.

Definition 2.3.3 (Simplicial Complex [67, Definition 2.3.5]). A *simplicial complex K* is a finite set of of *simplexes*, all contained in some Euclidean space R^m. Furthermore

1. if σ is a Simplex of K, and τ is a face of σ, then τ is in K.

2. if σ and τ are simplexes of K, then $\sigma \cap \tau$ either is empty, or is a common face of σ and τ.

An example of an arbitrary simplicial complex can be seen in Figure 2.4. A simplicial complex is called *empty*, if it consists solely of the empty simplex. The *void complex $\emptyset$* is the simplicial complex with no simplices in it.

Figure 2.4.: Example of arbitrary simplicial complex

Definition 2.3.4 (Deletion)**.** Let Σ be a simplicial complex and $\boldsymbol{\mu} \in \Sigma$ a simplex contained in Σ. Then

$$\Sigma - \boldsymbol{\mu} := \{\sigma \in \Sigma \mid \sigma \nsubseteq \boldsymbol{\mu}\}$$

Definition 2.3.5 (Maximal simplex, adapted from [33])**.** A simplex $\boldsymbol{\mu}$ of Γ is called a *maximal simplex* of Γ, iff it is not a proper subset of any other simplex of Γ. In the following $\max(\boldsymbol{\mu})$ will describe the set of maximal simplices of Γ.

$$\max(\Gamma) \ni \boldsymbol{\mu} \Leftrightarrow \nexists \gamma \in \Gamma \colon \boldsymbol{\mu} \subsetneq \gamma \,.$$

The term *facet* (not to be confuces with *face*) is used synonymously with the term maximal simplex.

Definition 2.3.6 (Free faces, adapted from [33])**.** A simplex $\gamma \in \Gamma$ of a simplicial complex Γ is called a *free face* of Γ, iff it is a proper subset of exactly one maximal simplex of Γ.

$$\Gamma \ni \gamma \text{ is free } \Leftrightarrow \exists! \boldsymbol{\mu} \in \max(\Gamma) \colon \gamma \subsetneq \boldsymbol{\mu} \,.$$

For annotation purposes we also define an operator, that returns true, if a given simplex is a free face on a specified simplicial complex, and the set of all free faces of a simplicial complex.

Definition 2.3.7 (isFree)**.** Let f be a simplex and S be a simplicial complex. The operator $\text{isFree}(f, S)$ is true, iff f is a free face of S.

$$\text{isFree}(f, S) \Leftrightarrow f \in S \wedge \exists! m \in \max(S) \colon f \subsetneq m \,.$$

Additionally, denote $\text{free}(S)$ the set of all free faces on S:

$$\text{free}(S) := \{f \mid \text{isFree}(f, S)\} \,.$$

Example 2.3.8. Consider the simplicial complex K pictured in Figure 2.5. K has three maximal complexes:

$$\{\{a,b,c\},\{b,d\},\{c,d\}\} = \max(K).$$

The faces $\{a,b,\},\{a,c\} \in \text{free}(K)$ are proper free faces, as they are proper subsets of maximal simplex $\Delta = \{a,b,c\}$. However $\{d\}$ is not a free face, as it is a proper subset of the two maximal simplexes $\{b,d\}$ and facet $\{c,d\}$.

Figure 2.5.: Simplicial complex K, consisting of a filled, and a hollow triangle

Definition 2.3.9 (Elementary collapse, based on [89][67, p. 72]). Given a simplicial complex K with a free face $\tau \in \text{free}(K)$. Let σ be the maximal simplex that contains τ: $\tau \subsetneq \sigma$. The process of passing from K to $K - \sigma - \tau$ is called an *elementary collapse*. The inverse operation is called an *elementary anti-collapse*. If there exists a series of elementary collapses, that transform K into the simplicial complex containing only a single vertex, K is called *collapsible*.

Definition 2.3.10 (Simple-homotopy equivalence [8, 89]). Two simplicial complexes are of the same *simple-homotopy type* (are simple-homotopy equivalent), if they can be transformed into one another via some sequence of collapses and anti-collapses.

We note, that simple-homotopy equivalence is a special case of homotopy-equivalence [89], which we will not define here, and instead refer to [71, p. 108] for a definition.

2.4. Model Formalization

In this section, we formalize the model on which the topological notions will work. The privacy-notions work on relations between attributes and individuals having these attributes. The relation between user and attribute is established when a user possesses a certain attribute. We usually represent these relations as tables, where columns represent binary attributes and rows individuals, and ● denotes the presence of an entry, an example of this can be seen in Table 2.2. Mathematically, we can formalize this as follows:

Definition 2.4.1 (Individual-Attribute-Relation). Let X be a finite discrete nonempty space of individuals, Y a finite discrete nonempty space of attributes and R a relation on $X \times Y$. Relation R is a set of ordered pairs (x,y) with $x \in X$ and $y \in Y$.

Row: For each $x \in X$ let $Y_x = \{y \in Y \mid (x,y) \in R\}$. Then Y_x consists of all attributes of individual x.

Column: For each $y \in Y$ let $X_y = \{x \in X \mid (x,y) \in R\}$. Then X_y consists of all individuals who have attribute y.

Given a relational database, we can define the Dowker Complexes.

Definition 2.4.2 (Dowker Complexes, adapted from [33]). Let R be a relation over individuals and attributes $X \times Y$. Then the *Dowker Complexes* are defined as follows.

$$\Phi_R = \{\gamma \subseteq Y \mid \forall y \in \gamma \exists x \in X : (x, y) \in R\}$$
$$\Psi_R = \{\sigma \subseteq X \mid \forall x \in \sigma \exists y \in Y : (x, y) \in R\}.$$

Note that for every relational database, the Dowker Complexes are simplicial complexes. These complexes Φ_R and Ψ_R, have the same homotopy type (are *homotopy equivalent*), and we say they are Dowker dual of each other with respect to relation R.

Thanks to this construction we obtain a topological model of our data, which allows us to transfer geometric properties to privacy ones. To do so, we need to understand the interpretation of these two complexes:

- Φ_R models attributes shared by individuals. A set of attributes $\gamma \neq \emptyset$ is in simplex Φ_R, iff there exists (at least) one individual x_i in the relation with a set of attributes γ_i with $\gamma \subseteq \gamma_i$. In this case x_i is called a witness for γ.

- While Ψ_R models individuals with shared attributes. A set of individuals $\sigma \neq \emptyset$ is in Ψ_R, iff there exists (at least) one attribute y_i in the relation which is shared by a group of individuals σ_i with $\sigma \subseteq \sigma_i$. y_i is called a witness for σ.

For readability, Φ_R will sometimes be referred to as the *attribute complex*, while Ψ_R will be referred to as the *association complex*.

Example 2.4.3. Consider the following relation NH which records for the participants Alice, Bob, Charlie and Dough, if they live in neighborhood A or B, if they have recently been to a hospital and if their income is under a certain threshold. Therefore, we have the following individuals and attributes sets respectively:

$$X = \{\text{ALICE, BOB, CHARLIE, DOUGH}\}$$
$$Y = \{\text{NEIGHBORA, NEIGHBORB, HOSPITAL, INCOME}\}.$$

The relation is defined via Table 2.2.

NH	NEIGHBORA	NEIGHBORB	HOSPITAL	INCOME
ALICE	•		•	•
BOB	•			
CHARLIE		•	•	
DOUGH		•		•

Table 2.2.: NH relation on $X \times Y$

The Dowker complexes for this relation are the attribute complex Φ_{NH}, and the association complex Ψ_{NH}. They are illustrated in Figure 2.6.

The attribute complex (Figure 2.6a) contains simplices made of sets of attributes. We can see that $\sigma = \{\text{NEIGHBORA, HOSPITAL, INCOME}\}$ is one such simplex. In this case ALICE is a witness for σ.

The association complex Ψ_{NH} (Figure 2.6b) contains sets of individuals. An example of a witness in the association complex is the NEIGHBORA that witnesses $\{\text{ALICE, BOB}\}$.

For completeness, the following lists all simplices in the attribute complex, and their corresponding witnesses:

(a) Φ_{NH} (b) Ψ_{NH}

Figure 2.6.: Dowker Complexes of NH

- {NEIGHBORA, HOSPITAL, INCOME}, witness: ALICE.

- {NEIGHBORA, HOSPITAL}, witness: ALICE.

- {NEIGHBORA, INCOME}, witness: ALICE.

- {HOSPITAL, INCOME}, witness: ALICE.

- {NEIGHBORB, HOSPITAL}, witness: CHARLIE.

- {NEIGHBORB, INCOME}, witness: DOUGH.

- {NEIGHBORA}, witness: ALICE or BOB.

- {NEIGHBORB}, witness: CHARLIE or DOUGH

- {HOSPITAL}, witness: ALICE or CHARLIE.

- {INCOME}, witness: ALICE or DOUGH.

And these are the simplices and corresponding witnesses of the association complex:

- {ALICE, BOB}, witness: NEIGHBORA.

- {CHARLIE, DOUGH}, witness: NEIGHBORB.

- {ALICE, CHARLIE}, witness: HOSPITAL.

- {ALICE, DOUGH}, witness: INCOME.

- {ALICE}, witness: NEIGHBORA, HOSPITAL or INCOME.

- {BOB}, witness: NEIGHBORA.

- {CHARLIE}, witness: NEIGHBORB or HOSPITAL.

- {DOUGH}, witness: NEIGHBORB or INCOME.

Definition 2.4.4 (Poset)**.** A *partially ordered set*, in short *poset* is a set of elements with a partial order ($\leq$)

Let R be a relation over individuals X and attributes Y, then we can define the poset P_R associated with the relation R (Adapted from [33]): Let the nonempty set $\emptyset \neq \sigma \subseteq X$ describe a set of individuals, and the nonempty set $\emptyset \neq \gamma \subseteq Y$ describe a set of attributes. A poset element of P_R consists of an ordered pair (σ, γ). Since, it is labeled with both σ and γ this kind of poset is also called *doubly labeled*.

For the poset elements and their order the following holds:

- All individuals in σ have all attributes in γ.

- Iff an individual has at least all attributes in γ, then that individual must be in σ.

- Iff an attribute is shared by at least all individuals in σ, then that attribute must be in γ.

- Partial order is described by edges in the graph.

- There is an edge between (σ_1, γ_1) and (σ_2, γ_2) iff corresponding sets are subset comparable.

- $(\sigma_1, \gamma_1) \leq (\sigma_2, \gamma_2)$, when $\sigma_1 \subseteq \sigma_2$ and $\gamma_1 \supseteq \gamma_2$. So the poset is ordered from attributes that are less shared to attributes that are shared by a lot of people

These posets are designed to model inference processes. Attribute inference: Given a set γ of attributes, if γ is subset of γ_i in P_R and γ_i is the smallest such superset, we know individuals with attributes γ also have attributes γ_i. This lets us infer new information if $\gamma \subsetneq \gamma_i$. Analogous for association inference with σ and σ_i.

Example 2.4.5. Continuing Example 2.4.3, we replace individuals names with corresponding numbers 1–4. The poset elements can be gathered from the simplicial complexes' (recall Figure 2.6) simplices and their witnesses. In order to illustrate this we give the following examples:

$$(\sigma_1, \gamma_1) =: (\{1\}, \{\text{NEIGHBORA}, \text{HOSPITAL}, \text{INCOME}\})$$

is an element of the poset since individual 1, Alice, is a witness for $\{\text{NEIGHBORA}, \text{HOSPITAL}, \text{INCOME}\}$ in Φ_{NH}. Another example is

$$(\sigma_2, \gamma_2) =: (\{1, 2\}, \{\text{NEIGHBORA}\})$$

since individuals 1 and 2 are witnesses for $\{\text{NEIGHBORA}\}$ in Φ_{NH}, or since NEIGHBORA is a witness for $\{1, 2\}$ in Ψ_{NH}. Furthermore, these elements are subset compatible, since

$$\{1\} = \sigma_1 \subseteq \sigma_2 = \{1, 2\}, \text{ and}$$
$$\{\text{NEIGHBORA}, \text{HOSPITAL}, \text{INCOME}\} = \gamma_1 \supseteq \gamma_2 = \{\text{NEIGHBORA}\}.$$

So $(\sigma_1, \gamma_1) \leq (\sigma_2, \gamma_2)$. In other words, the attribute set γ_1 is shared by less people (only Alice has this combination), while the set γ_2 is shared by more people (both Alice and Bob have this attribute).

The full poset is illustrated in Figure 2.7.

Figure 2.7.: Doubly labeled partially ordered set for relation NH

Concerning attribute inference: Say we have the attribute set $\gamma = \{$NEIGHBORA, HOSPITAL$\}$, then γ is subset of γ_1, and γ_1 is the smallest such superset. The new information learned is that if an individual lives in neighborhood A and has been to the hospital, their income is under the threshold.

On the other hand, if the attribute set is $\gamma = \{$NEIGHBORA$\}$, then γ_2 is the smallest set such that $\gamma \subseteq \gamma_2$, and thus no new information is learned.

Definition 2.4.6 (Face Poset [33])**.** Let Σ be a nonvoid simplicial complex. The partially ordered set $\mathfrak{F}(\Sigma)$ determined by Σ is called the *face poset* of Σ.

The elements of this poset are the nonempty simplices of Σ. Partially ordered by set inclusion.

In this case, the connection between the two homotopy equivalent simplicial complexes Φ_R and Ψ_R translates to two specific maps between the face posets of these complexes.

Definition 2.4.7 (Operators ϕ_R and ψ_R [33])**.** Let $R \in X \times Y$ be a relation over individuals X and attributes Y. The operators between the face posets of attribute complex Φ_R and association complex Ψ_R, mentioned above, are defined as follows:

$$\phi_R \colon \mathfrak{F}(\Psi_R) \to \mathfrak{F}(\Phi_R) \qquad\qquad \psi_R \colon \mathfrak{F}(\Phi_R) \to \mathfrak{F}(\Psi_R)$$
$$\sigma \mapsto \bigcap_{x \in \sigma} Y_x \qquad\qquad\qquad \gamma \mapsto \bigcap_{y \in \gamma} X_y \, .$$

Further $\phi_R(\emptyset) \coloneqq Y$ and $\psi_R(\emptyset) \coloneqq X$.

Recall, that Y_x is the set of all attributes of an individual x and X_y is the set of all individuals that share an attribute y. So when in the definition of ϕ_R we cut Y_x over all individuals in σ, that means that we get the shared attributes between all of those individuals:

- $\phi_R(X) = \gamma$ is the set of attributes γ, that is shared between all individuals in the relation.

$\vdots$

- $\phi_R(\{x_1, x_2\}) = \gamma'$ is the set of attributes γ', that individuals x_1 and x_2 have in common.

- $\phi_R(\{x\}) = Y_x$ is the set of attributes Y_x of an individual x.

- $\phi_R(\emptyset) = Y$ is the set of all possible attributes Y in the relation, this would even include attributes that no individual has.

And analogously in the definition of ψ_R we cut X_y over all attributes in γ. So we get all individuals that share all of the attributes in γ:

- $\psi_R(Y) = \sigma$ is the set of individuals σ, that each have all possible attributes in the relation.

$$\vdots$$

- $\psi_R(\{y_1, y_2\}) = \sigma'$ is the set of individuals σ', that have both attributes y_1 and y_2 in common.

- $\psi_R(\{y\}) = X_y$ is the set of individuals X_y that have attribute y.

- $\psi_R(\emptyset) = X$ is the set of all individuals X in the relation, this would even include individuals that have no attributes.

Chaining both operators, on a set of individuals or attributes, will always produce a superset of the input. This is because moving from a set of individuals σ that share a common set of attributes γ, to the set of attributes γ' that is common between those individuals, the latter set of attributes should include the former. Dually, when moving from a set of attributes γ, that is shared between some individuals σ, the set of individuals σ', that all have the given attributes, should be larger or equal to the initial set of individuals σ.

Lemma 2.4.8 ([33, Lemma 43]). *Let R be a relation on $X \times Y$.*

$$\forall \gamma \subseteq Y, \quad \gamma \subseteq (\phi_R \circ \psi_R)(\gamma).$$

Dual properties hold:

$$\forall \sigma \subseteq X, \quad \sigma \subseteq (\psi_R \circ \phi_R)(\sigma).$$

This means such a chaining of these operators maps a face of the corresponding Dowker complex, to a simplex that contains it. Consequently, a maximal face will always be mapped onto itself

Corollary 2.4.9 ([33, Corollary 44]). *Let R be a relation on $X \times Y$, with both X and Y nonempty. If γ is a maximal simplex of Φ_R, then $(\phi_R \circ \psi_R)(\gamma) = \gamma$.*
Dual properties hold: If σ is a maximal simplex of Ψ_R, then $(\psi_R \circ \phi_R)(\sigma) = \sigma$.

2.4.1. Assumptions of Attribute Privacy and Association Privacy

Before we finally define the topological notions of Attribute Privacy and Association Privacy, we present the assumptions for these notions, as they are presented by [33], and discuss implicitly held assumptions.

Relational Completeness: Any given relation is reflecting the ground truth fully. Any missing elements observed in reality will deem relation *inconsistent*. Unobservable but valid pairs can be omitted.

Observational Monotonicity: Observation of an attribute for an individual implies that this individual has that attribute. In other words, the observation of an attribute is meaningful.

However, the absence of attribute is not significant. If an attribute is not observed for an individual, it does not imply that the individual does not have that attribute. This means a lack of properties will not be protected.

So an implicit assumption is that the notion only needs to protect given information. It is however noted, that if the absence of an attribute is significant and observable, then that attribute should appear in its original and negated form in the relation. This assumption is used by [33] to view relationships as monotone boolean functions and thus enable the use of combinatorial topology.

Observational Accuracy: Observations are accurate. Relationship defines bipartite graph, assumptions describe (observations of) a sensor. It is however explicitly allowed to blur relationships (allow uncertainty) to preserve privacy.

Implicit Assumptions: Some assumptions are not explicitly stated, but can be inferred:

- A single record always represents one person (as opposed to the possibility of multiple people per record, or multiple records for a person).

- Presence or absence in database by itself is not sensitive: This follows from assuming, that only the presence of attributes is significant.

- Attribute Privacy implicitly assumes, that explicit identifiers have been removed.

- Association Privacy on the other hand, assumes, that explicit identifiers are present.

- No background knowledge abut correlations between individuals or attributes is assumed.

Strong Attacker: These notions do not assume any prefixed structure of background knowledge of the attacker.

- Attribute Privacy assumes, that the attacker knows a set of attributes, that the target individual holds (with the goal of uncovering more attributes of the target individual).

- Association Privacy assumes, that the attacker knows a set of individuals, that are associated with each other and can locate those individuals in the relation (with the goal of inferring further associations).

2.5. Topological Privacy Notions

Now that the necessary background has been given, we can define the topological privacy notions *Attribute Privacy* and *Association Privacy.*

Attribute Privacy aims to protect the attributes of an individual. It does so, by ensuring that knowing some set of attributes of an individual, the released information, does not let an attacker infer more information on that individual. And Association Privacy aims to conceal relationships such as "work on the same secret project together" between individuals in the released relation. It does so by ensuring, that an adversary, who knows

a set of individuals that are in some relationship with each other, cannot infer more individuals that are in the same relationship.

Both notions achieve their protection goals, by chaining the previously defined operators ψ_R and ϕ_R — that map a set of attributes to the set of corresponding individuals and a set of individuals to corresponding attributes respectively — and requiring the output to not include any further elements, as this would constitute an inference of attributes or associations respectively.

Definition 2.5.1 (Attribute Privacy [33])**.** A relation R with nonvoid Dowker complexes preserves *Attribute Privacy* iff

$$\phi_R \circ \psi_R \text{ is identity operator on } \mathfrak{F}(\Phi_R) \cup \{\emptyset\} \,.$$

So the operator $\phi_R \circ \psi_R$ can be interpreted as describing how the observation of some attributes implies the existence of other attributes (inference process). And Attribute Privacy can be interpreted as asking the question "If we go from a set of attributes γ to all individuals $x \in \sigma$ that share those attributes, and then go back to all attributes γ' that those individuals share, do we end up with the same set of attributes?"

In the case of a relation that does not preserve Attribute Privacy, this set γ' is a superset of γ and so if we have the background knowledge that the target individual has attributes γ we are able to infer from the relation that the target must also have attributes $\gamma \setminus \gamma'$.

This could work as follows: We know our target individual is in the relation R and has the attributes "works at company A" (COMPANYA), and "rides bike to work" (BIKE). We find out, that individuals 2, 5, and 12 all share these attributes, and in a next step we analyze, which (other) attributes are shared between those individuals:

$$\phi_R \circ \psi_R(\{\text{COMPANYA}, \text{BIKE}\}) = \phi_R(\{2, 5, 12\}) = \{\text{COMPANYA}, \text{BIKE}, \text{NEIGHBORA}\} \,.$$

This then reveals, that the target also must live in neighborhood A, thus privacy of attributes is not preserved.

The operator $\phi_R \circ \psi_R$ will always produce an output set that contains at least the input set, as all individuals that share a given attribute set, share that attribute set. The problem is that the output set could contain *additional* attributes. So for a relation to preserve Attribute Privacy, we require the operator $\phi_R \circ \psi_R$ to be the identity operator, as any other operator will reveal attributes.

Definition 2.5.2 (Association Privacy [33])**.** A relation R with nonvoid Dowker complexes preserves *Association Privacy* iff

$$\psi_R \circ \phi_R \text{ is identity operator on } \mathfrak{F}(\Psi_R) \cup \{\emptyset\} \,.$$

Similarly, $\psi_R \circ \phi_R$ can be interpreted as describing how associations between individuals imply other such associations. And Association Privacy can be interpreted as asking the question "If we go from a set of individuals σ to all attributes that those individuals share γ and then back to who (else, σ') has those attributes, do we end up with the same set of individuals we started with?"

If a relation does not preserve Association Privacy, there is a set of individuals σ for which we find individuals $y \notin \gamma$ that have all attributes σ that are shared between the individuals of γ as well. So we were able to associate these individuals with the individuals of γ.

The duality between the two notions can be seen in the following intuition: With a relation represented as a table, in which individuals each have a corresponding row, and attributes have a corresponding column, the Attribute Privacy operator $\phi_R \circ \psi_R$ maps between rows, and the Association Privacy operator $\psi_R \circ \phi_R$ maps between columns.

2.6. Properties and Theorems of Topological Notions

With the topological privacy notions defined, this section focuses on how to generally achieve them and points out some notable properties of the notions.

Definition 2.6.1 (Unique Identifiability, adapted from [33])**.** Let R be a relation R on $X \times Y$. An individual $x \in X$ of relation R is said to be *uniquely identifiable via relation R*, iff $\psi_R(Y_x) = \{x\}$. In other words, x is uniquely identifiable via R, when the set of all individuals, who also have all attributes that x has, only contains x.

Take for example a relation R with the entries $\{1 \mapsto \{a, b, c\}, 2 \mapsto \{b\}, 3 \mapsto \{c\}\}$. Neither individual 2, nor individual 3 are uniquely identifiable, because individual 1 has all of their attributes.

The absence of free faces in the attribute complex [association complex] imply attribute [association] privacy. And conversely, with the notable exception of individuals that are not uniquely identifiable, the presence of free faces in the attribute [association] complex, implies loss of attribute [association] privacy. This can be formalized as follows:

Lemma 2.6.2. *Let R be a relation on $X \times Y$, with X, Y being nonempty. Consider the following three statements*

1. *The attribute complex Φ_R contains no free faces (NO-FREE-FACE).*

2. *R preserves Attribute Privacy (ATTRIBUTE-PRIVACY).*

3. *Every individual on R is uniquely identifiable (ALL-UNIQUE)*

The following implications hold:

$$NO\text{-}FREE\text{-}FACE \Rightarrow ATTRIBUTE\text{-}PRIVACY$$

$$NO\text{-}FREE\text{-}FACE \Leftarrow ATTRIBUTE\text{-}PRIVACY \wedge ALL\text{-}UNIQUE$$

Dual properties to these statements also hold for Association Privacy.

This Lemma is proven in [33, Lemma 61–63].

To achieve either of the notions, in many cases, the original relation has to be altered. The types of alterations that can be made to a relation include expanding the dimension of the relation by adding new columns (attributes) or rows (individuals) but also modifying the existing relation. These types of alterations have been given names to enable a clear description of what is being done.

Definition 2.6.3 (Disinformation [33])**.** An artificial entry in a relation added to achieve privacy is called *disinformation*.

Definition 2.6.4 (Discontinuity [33])**.** Discarded knowledge (so a removed entry) in relation is called a *discontinuity*.

Disinformation and discontinuities skew statistics and utility but (aim to) increase Attribute and/ or Association Privacy

Attribute and Association Privacy at the same time   A further theorem states, that a relation with more attributes than individuals cannot preserve attribute privacy:

Theorem 2.6.5 (Too many attributes [33, Theorem 65])**.** *Let R be a relation on $X \times Y$ with no blank columns. Suppose that $|Y| > |X| \geq 1$. Then R does not preserve attribute privacy.*

The intuition to this theorem is, that for a given individual x that has an attribute a, there needs to be another individual x', that does not have this attribute. Individuals x and x' are allowed to share other attributes. Otherwise an attacker would be able to — given the set of shared attributes between x and x' (which can also be the empty set) — infer the attribute a from that set of shared attributes. This means, roughly speaking, each attribute needs at least one individual, that does not possess this attribute.

A dual theorem holds for Association Privacy. In [33, E.3] Erdmann proves that a relation that preserves both attribute and Association Privacy needs to have the same amount of individuals as records. Further, both corresponding Dowker complexes need to be homotopic to spheres. It is proven that these relations come in one of a few forms (Isometric to a cyclic staircase, to a spherical boundary relation or be composed of such relations). And while these possibilities are non-trivial, they do severely constrain the range of possible relations with both of these properties.

3. Additional Theorems

In this chapter, we present a novel characterization of topological notions using free faces. This theorem (Theorem 3.0.1) enhances the understanding of Attribute Privacy in correspondence with the attribute complex as well as will be crucial in the rest of the thesis to prove properties in terms of composition, it further simplifies other proofs, in which we need to show that a given relation has Attribute Privacy. Besides, we proportionate an additional general definition and lemma — about the union of multiple relations, and how that affects the attribute complex — which we also use for our composability arguments. We state these here, as these are general results, that could be applied in other use cases as well.

We introduce a generalization of Lemma 2.6.2.

Theorem 3.0.1. *Let R be a relation on individuals X and attributes Y. R has Attribute Privacy iff for each free face f on Φ_R, there exists an individual x^f that has the attribute set $Y_{x^f} = f$ exactly corresponding to that free face:*

$$\forall f \in \text{free}(\Phi_R) \exists x^f \in X : Y_{x^f} = f \Leftrightarrow R \text{ has Attribute Privacy}.$$

This means for any given free face f on Φ_R, there exists an individual x^f that is witness to f, but not a witness to any $m \supsetneq f$. By definition, $\exists m \in \max(\Phi_R) : f \subsetneq m$ and there must exist a witness to m. This is what makes x^f non-unique. Thus, this theorem is a generalization of Lemma 2.6.2, because if Φ_R has no free faces, the left part of the equation will always be true, and because the non-unique individuals of Lemma 2.6.2 are exactly the witnesses to the free faces.

Proof. "$\Rightarrow$": Assume $\forall f \in \text{free}(\Phi_R) \exists x^f \in X : Y_{x^f} = f$. We will show R has Attribute Privacy by showing, that

$$\forall \gamma \in \mathfrak{F}(\Phi_R) \cup \{\emptyset\} : \phi_R \circ \psi_R(\gamma) = \gamma.$$

If Φ_R has no free faces ($\text{free}(\Phi_R) = \emptyset$) we know by Lemma 2.6.2 that R has Attribute Privacy and we are done. So from now on we will assume that $\text{free}(\Phi_R) \neq \emptyset$. Each nonempty simplex of Φ_R can be categorized in one of three categories: Maximal simplices, free faces, and simplices that are a child of at least two maximal simplices.

We will treat the empty simplex separately. We then consider all remaining cases, by analyzing each possible category for γ separately.

- Case 1 ($\gamma = \emptyset$) This means that $\psi_R(\emptyset) = X$, so we have to show that $\phi_R(X) = \emptyset$. To do this we will show that $\exists x_1, x_2 \in X$ with $Y_{x_1} \cap Y_{x_2} = \emptyset$. Consider $f \in \text{free}(\Phi_R)$ and $m \in \max \Phi_R : f \subsetneq m$. By assumption $\exists x_1 : Y_{x_1} = f$. Next we are concerned with $\eta \coloneqq m \setminus f$.
    - Now either $\exists f' \subseteq \eta : f' \in \text{free}(\Phi_R)$, for which by assumption $\exists x_2 : Y_{x_2} = f'$.
    - Or $\exists m' \in \max(\Phi_R)$ with $m' \neq m$ and $m' \supsetneq \eta$. Because m' is maximal, there exists a witness for it $\exists x_2 : Y_{x_2} = m'$

In both cases there is no overlap between the chosen simplices $Y_{x_1} \cap Y_{x_2} = \emptyset$.

- Case 2 ($\gamma \in \max(\Phi_R)$) by Corollary 2.4.9, $\phi_R \circ \psi_R(\gamma) = \gamma$.

- Case 3 ($\gamma \in \mathrm{free}(\Phi_R)$) by the assumption $\exists x^f \in X : Y_{x^f} = \gamma$. Let $\psi_R(\gamma) =: \sigma$, then $x^f \in \sigma$. It follows that

$$\phi_R \circ \psi_R(\gamma) = \phi_R(\sigma) = \bigcap_{x \in \sigma} Y_x = \bigcap_{x \in \sigma \setminus \{x^f\}} Y_x \quad \cap Y_{x^f}$$

$$= \bigcap_{x \in \sigma \setminus \{x^f\}} Y_x \quad \cap \gamma \subseteq \gamma \overset{\mathrm{Lemma\ 2.4.8}}{\Longrightarrow} \phi_R \circ \psi_R(\gamma) = \gamma \,.$$

- Case 4 ($\gamma \notin \max(\Phi_R) \wedge \gamma \notin \mathrm{free}(\Phi_R)$). Let $\psi_R(\gamma) =: \sigma$. This means

$$\exists M = \{m \in \max(\Phi_R) \mid \gamma \subsetneq m\} \wedge |M| > 1$$

$$\phi_R \circ \psi_R(\gamma) = \bigcap_{x \in \sigma} Y_x \overset{\mathrm{maximality}}{\subseteq} \bigcap_{m \in M} m \overset{|M| > 1}{=} \gamma \,.$$

"$\Leftarrow$": Assume R has Attribute Privacy. Show that

$$\forall f \in \mathrm{free}(\Phi_R) \exists x^f \in X : Y_{x^f} = f \,,$$

using proof by contradiction. If there are no free faces there is noting to show. Assume there is at least one free face without a witness in the relation $\exists \in \mathrm{free}(\Phi_R) \forall x \in X : Y_x \neq f$. By definition $\exists m \in \max(\Phi_R) : f \subsetneq m$. Because m is maximal, there has to be a witness for m, and because f is free it is not encompassed by another maximal simplex.

$$\exists x^m \in X : Y_{x^m} = m$$
$$\nexists m' \in \max(\Phi_R) : m' \neq m \wedge f \subsetneq m' \,.$$

It follows that

$$\phi_R \circ \psi_R(f) = \phi_R(\{x^m\}) = Y_{x^m} = m \not\supseteq f \,,$$

which is a conflict with the assumption. $\qquad\square$

We assert, that, like for Lemma 2.6.2, a dual theorem for Association Privacy could be constructed. With this Theorem, we observe, that strategies to achieve the notions of Attribute Privacy or Association Privacy can involve the elimination of free faces.

We briefly give a privacy interpretation of this theorem: Each individual x in the relation is witness to the simplex $\gamma = Y_x$ in the attribute complex, representing all of their attributes. However, a given individual is also a witness to all simplices $\gamma' \subsetneq Y_x$ in the attribute complex.

Consider a set of attributes, that appears in the attribute complex as a simplex s that is the proper subset of two distinct maximal simplices m_1, m_2, that have only s in common. So an adversary that has the attribute set s as background knowledge cannot infer further attributes, as they cannot know if their target individual is a witness to m_1 or a witness to m_2.

Now consider a set of attributes, that appears in the attribute complex as a simplex f that is the proper subset of only one maximal simplex m. As m is maximal, there is a witness individual x, that produces m. Because $f \subset m$, individual x is also witness to f.

If there is no other witness x' to f, an adversary with f as background knowledge can infer that their target must also have attributes m. However, if there is another individual x', that is witness to f, but not witness to m, then an adversary cannot be sure if their target is x or x'.

Next, we define how two relations are combined, and analyze how this affects the attribute complex of the combined relation:

Definition 3.0.2 (Union of relations). Let X_A, X_B be nonempty sets of individuals, and Y_A, Y_B nonempty sets of attributes. Let R_A be a relation on $X_A \times Y_A$ and R_B be a relation on $X_B \times Y_B$. Then the union of those relations, is the relation R_{AB} on $(X_A \cup X_B) \times (Y_A \cup Y_B)$, which contains all entries of R_A and R_B.

$$R_{AB} = \{(x,y) \mid (x,y) \in R_A \vee (x,y) \in R_B\}$$

We observe, that if the input sets of attributes are the same, the attribute complex of the union of two relations, consists of the simplices of the source relations attribute complexes.

Lemma 3.0.3 (Union of Attribute Complexes). *Let X_A, X_B be nonempty sets of individuals, Y a nonempty set of attributes, R_A be a relation on $X_A \times Y$ and R_B be a relation on $X_B \times Y$. Let $R_{AB} = R_A \cup R_B$. Then*

$$\Phi_{R_{AB}} = \Phi_{R_A} \cup \Phi_{R_B}.$$

Proof.

$$
\begin{aligned}
\Phi_{R_{AB}} \;&\overset{\text{Definition 2.4.2}}{=} \{\gamma \subseteq Y \mid \forall y \in \gamma \exists x \in (X_A \cup X_B)\colon (x,y) \in R_{AB}\} \\
&= \{\gamma \subseteq Y \mid \forall y \in \gamma \exists x \in (X_A \cup X_B)\colon (x,y) \in (R_A \cup R_B)\} \\
&\overset{\text{Definition 3.0.2}}{=} \{\gamma \subseteq Y \mid \forall y \in \gamma \\
&\qquad\qquad \exists x \in (X_A \cup X_B)\colon (x,y) \in \{(x,y) \mid (x,y) \in R_A \vee (x,y) \in R_B\}\} \\
&= \{\gamma \subseteq Y \mid \forall y \in \gamma \\
&\qquad\qquad (\exists x \in X_A\colon (x,y) \in R_A) \\
&\qquad\qquad \vee \\
&\qquad\qquad (\exists x \in X_B\colon (x,y) \in R_B\} \\
&= \{\gamma \subseteq Y \mid \forall y \in \gamma \exists x \in X_A\colon (x,y) \in R_A\} \\
&\qquad\qquad \cup \\
&\qquad\qquad \{\gamma \subseteq Y/ \mid \forall y \in \gamma \exists x \in X_B\colon (x,y) \in R_B\} \\
&\overset{\text{Definition 2.4.2}}{=} \Phi_{R_A} \cup \Phi_{R_B}.
\end{aligned}
$$

$\square$

4. Privacy for Relational Data

In this chapter, we will compare the topological notions to their closer relatives: SDC notions on relational data. Relational data, not to be confused with relations, describes traditional databases, which can be represented as tables. This stands in contrast with graph-data, which we will discuss in chapter 5.

This chapter is structured in three sections. First we analyze the composability properties of Attribute Privacy (section 4.1). Next, in section 4.2, we compare the topological notions with the state-of-the-art for relational data. This systematization includes a comparison of assumptions of the notions, analysis of the relation the notions stand in with each other, and an evaluation of common attacks against the topological notions. Finally, in section 4.3 we discus, whether Attribute Privacy could improve on DP, when the input data is correlated.

4.1. Composability Analysis

In this Subsection we investigate the composition properties of Attribute Privacy. Namely we will investigate, whether Attribute Privacy composes in parallel, or sequentially

We would like to point out, that [33, Chapter 10] introduces the concept of *informative attribute release sequences*. While this concept also tackles sequentially released information, it is quite separate from the one of composability. Informative attribute release sequences work on the assumption, that some relation without Attribute Privacy has been released. And that an individual wants to disclose their attributes, one-by-one, while not being uniquely identified on the relation for the longest amount of time.

We observe that the topological notions do not quantify the "amount of privacy" (like the privacy budget ε in ε-DP) and can only be fulfilled or not fulfilled. It cannot be said that the privacy is broken by some degree, but rather can be broken entirely by new information.

Since Attribute Privacy is a syntactic notion, we want to focus on released relations that both have Attribute Privacy, and not concern ourselves with the mechanism used to create those releases. However it is of note, that for sequential composition, it can happen that the same pair of individual and attribute (a cell in the tabular representation) appear in both releases, and contain conflicting data. This additional information might be used by an attacker, to infer some information about the "true" state of the original database, from which these releases were created. This cannot be modeled in the context of Attribute Privacy. If we avoid these conflicts, we can avoid discussions about original data and mechanisms entirely and only concern ourselves with the released relations themselves. We note, that in parallel composition there is no overlap by definition so these kinds of conflicts do not occur.

For sequential composition we formalize this notion of non-conflicting releases:

Definition 4.1.1 (Static and Dynamic Releases)**.** Let

- X be a set of individuals, and Y a set of attributes.

- R be a relation on $X \times Y$.

- $X_A, X_B \subseteq X$ with $X_A \cap X_B \neq \emptyset$ be two overlapping subsets of individuals.

- $Y_A, Y_B \subseteq Y$ subsets of attributes.

- $R_A \subseteq R, \quad R_A \in X_A \times Y_A$ a sub relation of R:

$$R_A \coloneqq \{(a,b) \mid (a,b) \in R \wedge a \in X_A \wedge b \in Y_A\}.$$

- $R_B \subseteq R, \quad R_B \in X_B \times Y_B$ a sub relation of R:

$$R_B \coloneqq \{(a,b) \mid (a,b) \in R \wedge a \in X_B \wedge b \in Y_B\}.$$

- $\mathcal{M} \colon V \times W, \ \text{aux} \mapsto V \times W$ be a release mechanism for relations ensuring Attribute Privacy, where aux is an arbitrary auxiliary input.

Further let $\mathcal{M}(R_A, \emptyset) \coloneqq R'_A$ be a release of R_A and $\mathcal{M}(R_B, R_A) \coloneqq R'_B$ a release of R_B. The releases R'_A and R'_B are called *static*, iff

$$\forall x \in X_A \forall y \in Y_A \colon (x,y) \in R_A \Leftrightarrow (x,y) \in R_B. \tag{4.1.1}$$

Otherwise they are called *dynamic* releases.

So in static releases the new release should have the same data as the prior-release, save for the additional data.

In the following we will define parallel and sequential composition for Attribute Privacy, by requiring that the combination two (parallel or sequentially) released attribute private relations is also attribute private

4.1.1. Parallel Composition

We start with parallel composition for Attribute Privacy and show that Attribute Privacy always composes in parallel.

Definition 4.1.2 (Parallel Composition for Arribute Privacy)**.** Two attribute-private relations, R_A, R_B with the same set of attributes Y, but non-overlapping individuals $X \cap X' = \emptyset$ *compose in parallel*, iff the union of both relations $R_A \cup R_B$ is an attribute-private relation.

Proposition 4.1.3 (Attribute Privacy composes in parallel)**.** *Two attribute-private relations, R_A, R_B with the same set of attributes Y, but non-overlapping individuals $X \cap X' = \emptyset$, always compose in parallel.*

Proof. We denote by R_{AB} the union of relations $R_A \cup R_B$ (Definition 3.0.2). An we denote the corresponding attribute complexes by A, B and AB, respectively. Per Lemma 3.0.3, the attribute complex AB consists of all the simplices in A and B, $AB = A \cup B$.

We need to prove that R_{AB} satisfies Attribute Privacy.

First we show that no new free faces are created, so any free face in AB stems from either A or B. So we show that

$$\text{free}(AB) = \text{free}(A) \cup \text{free}(B). \tag{4.1.2}$$

Recall that for any given $f \in \text{free}(f, AB) \Rightarrow \exists m \in \max{(AB)} \colon f \subsetneq m$. Since no new simplices are created, f has to be a simplex in either A or B: $f \in AB \Rightarrow f \in A \vee f \in B$. Likewise and because no simplices are destroyed[1]

$$m \in \max{(AB)} \Rightarrow m \in \max{(A)} \vee m \in \max(B). \tag{4.1.3}$$

Now we consider all cases

- Case 1 ($f \in A \wedge f \in B$) $f \in A \wedge f \in B \overset{(4.1.3)}{\Longrightarrow} f \in (\text{free}(A) \cup \text{free}(B))$. If f appears in both source complexes, since m is maximal in at least one of the source complexes, f is free in at least one of the source complexes.

- Case 2 ($f \in A \wedge f \notin B$)
  - Case 2.1 ($m \in \max(A) \wedge m \notin \max(B)$) $f \in A \wedge m \in \max(A) \Rightarrow f \in \text{free}(A)$
  - Case 2.2 ($m \notin \max(A) \wedge m \in \max(B)$) Case not possible because

  $$m \in \max(B) \overset{\text{(B is SC)}}{\Longrightarrow} \forall s \subsetneq m \colon s \in B$$

  conflicts with assumptions $f \notin B \wedge f \subsetneq m$.
  - Case 2.3 ($m \in \max(A) \wedge m \in \max(B)$) Case not possible analogous to case 2.2.

- Case 3 ($f \in B \wedge f \notin A$) analogous to case 2.

With this we can now show the main part of the proof. We do this by analyzing two cases.

- Case 1 (($\text{free}(A) \cup \text{free}(B)) = \emptyset$) attribute complexes A and B have no free faces. We show that in this case AB does not have any free faces either and thus fulfils Attribute Privacy. We proceed by *reductio ad absurdum*:

  Assume AB has a free face f. Because of Equation 4.1.2 it follows that $f \in (\text{free}(A) \cup \text{free}(B))$ This stands in conflict the the case considered.

- Case 2 ($\text{free}(A) \cup \text{free}(B)) \neq \emptyset$) Now for the case in which one or both of the contributing simplicial complexes have free faces:

  Because of Equation 4.1.2 we know that any free face in AB is also a free face of A or B. Because of Theorem 3.0.1 we know that because R_A and R_B are attribute private, for each free face in A and for each free face in B there is a witness. Together this means that for each free face in AB there is a witness, thus R_{AB} has Attribute Privacy. Formally:

$$\text{free}(AB) = \text{free}(A) \cup \text{free}(B) \text{ (4.1.2)}$$
$$\wedge \qquad \forall f \in \text{free}(A)\ \exists x^f \in X_A \colon Y_{xf} = f \text{ (Theorem 3.0.1)}$$
$$\wedge \qquad \forall f \in \text{free}(B)\ \exists x^f \in X_B \colon Y_{xf} = f \text{ (Theorem 3.0.1)}$$
$$\Longrightarrow \qquad \forall f \in \text{free}(AB)\ \exists x^f \in X_{AB} \colon Y_{xf} = f$$
$$\overset{\text{Theorem 3.0.1}}{\Longrightarrow} \quad R_{AB} \text{ has Attribute Privacy}.$$

Thus we have shown, that in all possible cases R_{AB} has Attribute Privacy. $\qquad \square$

---

[1]m has to be maximal in the source simplicial complex it appears because $m \in X, X \in \{A, B\} \wedge m \notin \max(X) \Rightarrow \exists w \in X \colon m \subsetneq w \Rightarrow w \in AB \notin m \in \max(AB)$.

4.1.2. Sequential Composition

Next, we define sequential composition for Attribute Privacy and then show that Attribute Privacy does not generally compose sequentially.

Definition 4.1.4 (Sequential Composition for Attribute Privacy (Static Releases))**.** Let relations and releases be like in Definition 4.1.1. When R'_A and R'_B are static releases, they compose sequentially, iff $R'_A \cup R'_B$ has Attribute Privacy

Proposition 4.1.5. *Given two attribute-private relations, R_A, R_B with the same set of individuals X, but non-overlapping attributes $Y \cap Y' = \emptyset$, the union of both relations $R_A \cup R_B$ is not necessarily an attribute-private relation.*

The intuition behind this is that a relation with Attribute Privacy necessarily has more records than attributes, and by composing two databases side-by-side, can result in a database that has more attributes than records, even if the individual databases had Attribute Privacy.

Proof. We proof this by counterexample. Consider relation R, shown in Table 4.1a. If the sub-relation consisting of record 1 and attribute a is released, it has Attribute Privacy. Likewise, release the relation consisting of record 1 and attribute b and has it Attribute Privacy. However combining those back to relation R it is obvious that it does not have Attribute Privacy.

$$\{1\} \times \{a\} \ni R' := \{(1,a)\} \text{ has attr. privacy}$$
$$\{1\} \times \{b\} \ni R'' := \{(1,b)\} \text{ has attr. privacy}$$
$$\{1\} \times \{a,b\} \ni R = R' \cup R'' = \{(1,a),(1,b)\} \text{ does not have attr. privacy}.$$

$\square$

Further: Allowing the attribute sets to overlap, does not "solve" this issue.

Proposition 4.1.6. *Given two attribute-private relations, R_A, R_B with the same set of individuals X, and overlapping sets of attributes $Y \neq Y' \wedge Y \cap Y' \neq \emptyset$, the union of both relations $R_A \cup R_B$ is not necessarily an attribute-private relation.*

Proof. Consider relation S, shown in Table 4.1b. If the sub-relation consisting of records $1, 2$ and attributes a and b is released, it has Attribute Privacy. Likewise release the relation consisting of records $1, 2$ and attributes b and c and it has Attribute Privacy. However combining those back to relation S it is obvious that it does not have Attribute Privacy.

$$\{1,2\} \times \{a,b\} \ni S' := \{(1,a),(2,b)\} \text{ has attr. privacy}$$
$$\{1,2\} \times \{b,c\} \ni S'' := \{(2,b)(1,c)\} \text{ has attr. privacy}$$
$$\{1,2\} \times \{a,b,c\} \ni S = S' \cup S'' = \{(1,a),(2,b),(1,c)\} \text{ does not have attr. privacy}.$$

$\square$

R	a	b
1	●	●

S	a	b	c
1	●		●
2		●	

(a) Tiny relation that does not compose

(b) Small relation that does not compose

Table 4.1.: Small relations that do not compose sequentially

4.1.3. Attribute Privacy under Continual observations

A common problem in SDC is to provide repeated private access to data in the scenario in which data needs to be updated over time. This is called *privacy under continual observations* of a system's state [24, 29]. This is broadly studied in SDC, because it has numerous applications, such as monitoring traffic conditions, search trends, or epidemic incidences [29].

While we showed in subsection 4.1.2, that generally two releases do not compose sequentially, it is possible to construct a mechanism that produces static releases that do compose sequentially. For example the mechanism that always produces the empty relation for the second release. Of course such a mechanism would not be useful in any practical application. So here we want to analyze if it is possible to construct a *useful* mechanism for continual observation. Assume R is a relation over individuals X and attributes $Y = \{a_1, \ldots, a_n\}$. To match the use case this mechanism would produce a sequence of static releases $R_1 \subsetneq \ldots \subsetneq R_n$ with $R_i = R_{i-1} \cup \delta_i, \in [1..n]$ and $\delta_i \in X \times \{a_i\}$. To satisfy sequential composition, R_{i-1} and R_i have to sequentially compose for $i \in [2..n]$. In the following we motivate, why we believe that such a mechanism would provide poor utility.

We would like to note that relations with more attributes than individuals cannot have Attribute Privacy, and thus with more attributes and the same amount of individuals, achieving Attribute Privacy gets "harder". As more attributes are added the likelihood of Attribute Privacy being broken increases, and more disinformation or discontinuities are being required.

Assume the mechanism has already released R_{j-1} and the released data contains an individual x that has an attribute a_i and no other individual has that attribute. If now a new attribute a_j is introduced and individual x would have this attribute, the release cannot contain it, otherwise $\phi \circ \psi(\{i\}) = \{i, j\}$. In any subsequent release R_k, individual x cannot contain a_k either for the same reason. The same logic can be generalized, and any individual that has is unique in any release. It cannot gain any new attributes in any new releases without loosing Attribute Privacy.

We conclude that Attribute Privacy is likely not suitable for continual observation.

4.2. Analysis and Systematization

In this section, we will compare the topological privacy notions with current state-of-the-art privacy notions, in their assumptions and attack resilience. This includes an analysis of possible relationships between the notions.

4.2.1. Comparison of Assumptions

In the following, we discuss the differences between the (explicit and implicit) assumptions of k-Anonymity, ε-Differential Privacy, and the topological notions.

All notions discussed here, except for Association Privacy, (implicitly) assume that explicit identifiers have been removed from the database. In contrast, Association Privacy, assumes individuals and their properties are known.

An assumption that is shared by all notions discussed here, without exception, is that each record represents a single distinct individual. Note however that this is an assumption for the underlying data, and not necessarily for the anonymized dataset, in which records may be duplicated. Both of these common assumptions are fairly standard, and representative of many scenarios, in which data about individuals is stored and shared.

k-Anonymity assumes that a single quasi-identifier QID that contains all attributes that can be used to identify a person within a database exists and is known to the data publisher [37]. This also means that any other combination of attributes cannot be used to identify an individual within a database. The topological notions, or ε-Differential Privacy do not share this assumption. Assuming a known QID is the biggest drawback of k-Anonymity, since it relies on the publisher getting it right at the time of publishing, which is not necessarily very realistic.

The topological notions assume that the presence of an individual in the database in itself is not sensitive (only the presence of attributes is significant). k-Anonymity does not make any explicit assumptions about this. This stands in contrast to ε-Differential Privacy, which assumes the presence (or the absence) of the target's record in the database already reveals some sensitive information about the target [27]. This is a realistic assumption, especially in cases, in which the mere participation in a study holds a certain social stigma, f.e. a medical study on patients with a sexually transmitted disease.

Additionally ε-Differential Privacy assumes, that aside from the record of the target individual, the adversary knows all records of the input database. This assumption assures strong protection, and protects against the case, in which multiple malicious individuals in the database collude. Neither the topological notions nor k-Anonymity share this assumption.

Attribute Privacy's and Association Privacy's assumption of observational monotonicity and the accompanying implicit assumption that the lack of a property is not significant and only given information needs to be protected, is not shared by either k-Anonymity or ε-Differential Privacy. This assumption is coupled with the statement, that if the absence of a certain attribute should be important, then a negated version of that attribute should be introduced, which makes it more realistic, and avoids pathological counter examples. Even so, it is not a very common assumption to hold on relational data. However, unlike the QID-assumption of k-Anonymity, it does not have immediate obvious drawbacks on the privacy provided once it does not hold. Additionally [33] states, that it would be possible to give an alternative definition for the topological notions without this assumption.

None of the notions discussed here assume correlated data, which means they implicitly assume that there are no correlations between records or between attributes known to the adversary. In the context of ε-Differential Privacy, this is a topic of research, and alternative notions, or alterations have been proposed to solve this issue [95].

With these assumptions alone, we see, that ε-Differential Privacy provides by far the strongest privacy guarantees. In contrast, the assumptions of k-Anonymityweaken it

in a major way, and the topological notions likely stand in some place between. We summarized the assumptions of the notions in Table 4.2.

	Explicit identifiers removed	Record represents single distinct individual	Mere presence is sensitive	No QID	No obser-vational mono-tonicity	Correlated data
Attribute Privacy	✓	✓	✗	✓	✗	✗
Association Privacy	✗	✓	✗	✓	✗	✗
k-Anonymity	✓	✓	✗	✗	✓	✗
ε-Differential Privacy	✓	✓	✓	✓	✓	✗

Table 4.2.: Comparison of Assumptions between topological notions, k-Anonymity, and ε-Differential Privacy

4.2.2. Relationships between Notions

Sometimes different privacy notions are equivalent, proportional or simply one implies the other. This is the case of ℓ-Diversity that by definition implies k-Anonymity, in the sense that if a database satisfies the first condition it also satisfies the second one. In this section, we would like to compare the topological notions with the already existing ones in this perspective.

Since Topological notions are defined from a syntactic perspective, we will focus on the relation with the syntactic notions, in particular, with k-Anonymity. Besides, we first focus on implications on real data, so if presumed real database D fulfills notion $\mathcal{N}_1$, does it also fulfill notion $\mathcal{N}_2$?

To properly answer this question in relation to k-Anonymity we need first to clarify how to interpret the concept of quasi-identifier in relational databases. Observe that, k-Anonymity works on databases where attributes cannot only be binary, but enumerated as well, like "job", or "age". In contrast to that, the attributes in the relations, on which Attribute Privacy and Association Privacy work, can only be binary, like "smokes", or "isLawyer". So to facilitate the comparison assume data has the following form: For each non-binary attribute of the database, sub-attributes are introduced, that list all possible options for that attribute. To avoid confusion, when talking about attributes in a relation, those original attributes will be called attribute-groups and the options for these attribute-groups are called sub-attributes.

Definition 4.2.1 (Attribute-Group, Sub-Attributes)**.** Given a relational database D, providing values for attributes Y^D of individuals X. We call the attributes $y^D \in Y^D$ of the database *attribute-groups*. The database D can be transformed into a relation R over the same set of individuals X and a new set of binary attributes Y^R. We do this, by introducing a set of new attributes $\gamma \subseteq \text{Range}(y^D)$ for each $y^D \in Y^D$, representing possible values or value ranges of y^D. We call these new attributes *sub-attributes* We denote the transformation, that maps an attribute-group to a set of sub-attributes as a function DISCR. To note that the set of sub-attributes γ is created from attribute-group y^D, we write $\text{DISCR}(y^D) = \gamma$.

Relation R is created as a relation over all sub-attributes Y^R and individuals X, where

$$Y^R := \bigcup_{y^D \in Y^D} \text{DISCR}(y^D)\,.$$

For a given $y^D \in Y^D$, and a $y^R \in \text{DISCR}(y^D)$, we denote $D[x, y^D] = v \approx y^R$ to mean, that individual x has a value v in attribute y^D matching y^R.

The corresponding relation $R \in X \times Y^R$ then contains an individual-sub-attribute pair, if that individual contains a corresponding value in database D. Formally, for all $x \in X, y^D \in Y^D$, and all $y^R \in \text{DISCR}(y^D)$:

$$R := \left\{ (x, y^R) \colon D[x, y^D] \approx y^R \right\}$$

A general database that has been made compatible with Attribute Privacy and Association Privacy this way is shown in Table 4.3.

	a			b			c		$\cdots$
a_1	$\cdots$	a_l	b_1	$\cdots$	b_m	c_1	$\cdots$	c_n	$\cdots$

Table 4.3.: Compatible relationship, where a, b, c represent attribute-groups and a_i, b_i, c_i sub-attributes.

In the following we will base our discussions on the properties of relations, that were based on a source database (and thus attributes are sub-attributes, that belong to attribute-groups), but we do not discuss the properties of the source database itself.

Given a relation R, based on a database D, we need to understand what a set of quasi-identifier means: In this model a quasi-identifier is a set of attribute-groups. If we have access to the present sub-attributes of each attribute-group in the quasi-identifier of a user, we can identify that user. To clarify this we present the following example:

Example 4.2.2. We consider that the Job and Age form a quasi-identifier while the the cancer type is the sensitive attribute-group. User 1 is uniquely identifiable in this table since their entry is the only one with the quasi-identifier value $\gamma = \{Engineer, 31 - 49\}$. From this it can also be inferred that a record cannot have multiple sub-attributes of the

		Job			Age-Range			Cancer-Type		
	Lawyer	Engineer	Developer	Carpenter	≤ 30	31–49	≥ 50	Lung	Skin	Blood
1		•				•				•
2		•			•					•
3	•						•	•		
4			•		•					•

Table 4.4.: An example relational database following Table 4.3

same attribute-group. For example in Table 4.4, an entry cannot be in age-range ≤ 30 and 31–49 at the same time.

k-**Anonymity Vs. Association Privacy** With this context, considering that some set of attributes group considered a quasi-identifier we obtain that Association Privacy and *k*-Anonymity are mutually exclusive:

Proposition 4.2.3. *If a relationR fulfills k-Anonymity, then it does not fulfill Association Privacy and vice versa.*

In the following proofs k-ANON, ATTR, and ASSOC will refer to the variable of the relation fulfilling the respective notion.

Proof. (k-ANON $\Rightarrow$ ¬ASSOC) Let X be the user set corresponding to relation R. We choose an arbitrary user $x \in X$, and we denote by $\gamma \subseteq Y_x$ the attributes of x that correspond to a quasi-identifier value. Since R satisfies k-Anonymity, this record will be member of an anonymity group of size k. By definition all members of that anonymity group will share the same values for the quasi-identifier, therefore there exist at least $k - 1$ users $\{x_2, \ldots, x_k\}$ such that $\sigma = \{x, x_2, \ldots, x_k\} \subseteq \psi_R(\gamma)$.

This means that $\{x\} \neq \sigma \subseteq \psi_R \circ \phi_R(\{x\})$ and consequently we do not have Association Privacy.

ASSOC $\Rightarrow$ ¬k-ANON states the same as the above:

$$(k\text{-ANON} \Rightarrow \neg\text{ASSOC}) \Leftrightarrow (\neg k\text{-ANON} \vee \neg\text{ASSOC}) \Leftrightarrow (\text{ASSOC} \Rightarrow \neg k\text{-ANON}) . \qquad \square$$

k-Anonymity Vs. Attribute Privacy   Thanks to the previous discussion we can show that:

Proposition 4.2.4. *If a relatio R has Attribute Privacy (ATTR), it does not necessarily mean that the relation also fulfills k-Anonymity:*

$$ATTR \nRightarrow k\text{-ANON}$$

Proof. It is possible for a relation to have both Association Privacy and Attribute Privacy [33, E.3].

- Assume Attribute Privacy implies k-Anonymity, so any relation that has Attribute Privacy, also has k-Anonymity ($\forall$R ATTR $\Rightarrow$ k-ANON).

- Take relation R which has both Attribute Privacy and Association Privacy (ATTR $\wedge$ ASSOC).

- Then because R has Attribute Privacy, following the assumption, it also has k-Anonymity (ATTR $\wedge$ ASSOC $\overset{\text{ATTR}}{\Rightarrow}$ k-ANON),

- but also because it has Association Privacy, it cannot have k-Anonymity (ATTR $\wedge$ ASSOC $\overset{\text{ASSOC}}{\Rightarrow}$ ¬k-ANON$\nleftrightarrow$)

It follows that ATTR $\nRightarrow$ k-ANON. $\qquad \square$

t-Closeness and ℓ-Diversity both are defined over anonymity groups of k-Anonymity, and thus have the same comparability problem.

However, it is worth it to analyze in which cases we can have Attribute Privacy without k-Anonymity:

Proposition 4.2.5. *If a relation $R \in X \times Y$ fulfills Attribute Privacy and does not fulfill 2-Anonymity then all non 2-Anonymous individuals have none sensitive attributes.*

Proof. For all users $x \in X$ we can divide their attributes in the set of quasi-identifier value γ_1 and the set of sensitive attributes γ_2, this two sets of attributes are disjoint and since their union is the total information about x we have that $\phi_R(\{x\}) = \gamma_1 \cup \gamma_2$.

By hypothesis, R does not fulfill 2-Anonymity, therefore it exist an user x such that it is the only user with the quasi-identifier value γ_1, formally, $\psi_R(\gamma_1) = \{x\}$. By applying that, by hypothesis, R has Attribute Privacy, we have that

$$\gamma_1 \cup \gamma_2 = \phi_R(\{x\}) = \phi_R(\psi_R(\gamma_1)) = \gamma_1 \,.$$

Since γ_1, γ_2 are disjoint this means that $\gamma_2 = \emptyset$ finishing the proof. □

Until here we have explored the relation between databases fulfilling the notions. However, usually the real database does not directly fulfill the privacy notion and we need to use an anonymization mechanism to protect it. For this reason, we would like to analyze these notions from a more semantic perspective, comparing the results of mechanism whose outputs satisfy the notions.

4.2.2.1. Comparing the notions over Anonymized data

The following analysis focuses on a given mechanism $\mathcal{M}$, that works on presumed real data R and achieves a resulting database D which fulfills notion $\mathcal{N}_1$ (or on mechanism $\mathcal{M}$ that fulfills semantic notion $\mathcal{N}_1$). And determines if the resulting database (or the mechanism itself) also fulfills (or gives the same protections as) notion $\mathcal{N}_2$ on R. If that is the case we will say $\mathcal{N}_1$ *weakly implies* $\mathcal{N}_2$.

In this setting, k-Anonymity and Association Privacy are not mutually exclusive anymore, since the mechanism could be composed of two mechanisms, which first achieved k-Anonymity and then Association Privacy, or vice versa. However for the same reasons as previously discussed, they do not weakly imply each other either.

The interesting question is, in which relation k-Anonymity and Attribute Privacy stand.

Proposition 4.2.6. *k-Anonymity, does not weakly imply Attribute Privacy.*

Proof. At least two attributes are considered part of the quasi identifier, as explicit identifiers have been removed, and k-Anonymity only gives guarantees for quasi-identifiers. Name these attributes a_1 and a_2. Take a relation R in which at least one record x has the two attributes, so $\{(x, a_1), (x, a_2)\} \in R$. And let no entry have a_1 without having a_2 too, $\forall (x', a_1) \in R : (x', a_2) \in R$. This relation does not have Attribute Privacy, because $\{a_1, a_2\} = \phi_R \circ \psi_R(\{a_1\})$.

A mechanism can achieve k-Anonymity, without creating entries that have a quasi identifier that was not previously in the database. Such a mechanism will add enough fake entries that have both a_1 and a_2, and possibly other entries as well, but no entries that have a_1 without having a_2.

Thus the resulting database will thus still exhibit attribute leakage that corresponds to the real attributes of R, and not be attribute private. □

To analyze whether Attribute Privacy implies k-Anonymity, the assumptions under which the notions are achieved become important.

Proposition 4.2.7. *If operating sole under the assumptions of k-Anonymity, and explicitly going against the assumption of Attribute Privacy, that the absence of a relation entry does not imply an absence of that property for that person, Attribute Privacy does not weakly imply k-Anonymity.*

Proof. Consider Table 4.8 and relation Attr. This relation fulfills Attribute Privacy, we show this in Example 4.2.15. Let this relation be the output of some mechanism achieving Attribute Privacy, where records 2–4 are the real data, and all other records were added by the mechanism. Further let, attributes a, b the quasi-identifier. Then it is easy to see, that record 3 is the only record in its anonymity group. □

So for further analysis we will focus on k-Anonymity under the union of assumptions of k-Anonymity and Attribute Privacy. In particular, that the absence of an entry in the relation does not imply the absence of that attribute for that individual in the real world. Just that it has not been observed.

Proposition 4.2.8. *Under the unified assumptions of k-Anonymity and attribute-privacy, a given mechanism whose output always achieves Attribute Privacy, also always achieves k-Anonymity with $k = 2$.*

Proof. Let $R \in X \times Y$ be a relation of real data and $R' \in X' \times Y'$ the anonymized relation, as described above.

Let a subset of attribute-groups be the quasi-identifier.

Considering a record x of R', without loss of generality, this record has some present sub-attributes that are part of the quasi-identifier. Otherwise according to the assumptions that non-observations are not meaningful, that record would not show any persons presence in that relation. For the same reasoning, that record it has some sensitive sub-attributes. Else that record would only show the persons presence in the table and this is assumed to be not needed to be protected.

Should this records quasi-identifier γ_{QID} appear in R as well and have some present sub-attributes γ_s that are part of the sensitive attribute-groups, then that same quasi-identifier, also has to appear in R without said sensitive sub-attributes, as otherwise $\phi_R \circ \psi_R(\gamma_{\text{QID}}) = \gamma_{\text{QID}} \cup \gamma_s$.

Should that records quasi-identifier appear in R but without any sensitive attributes, then that would only show the presence of that record in the database, which is assumed to be non-sensitive. Any new quasi-identifiers in R do not have to be regarded, as they are not part of the original dataset.

So a mechanism $\mathcal{M}$ achieving Attribute Privacy for some relation $R \in X \times Y$, also achieves k-Anonymity, with $k \geq 2$.

However consider a mechanism $\mathcal{M}'$ which first executes $\mathcal{M}$ and then for a quasi-identifier γ_{QID} in R removes all but two records with this quasi-identifier. $\mathcal{M}'$ would still achieve Attribute Privacy, as long as the remaining records with the quasi-identifier γ_{QID} are the record that just has γ_{QID} and the record with has exactly one additional sensitive sub-attribute. It is clear, that the resulting relational database would not achieve $k \geq 3$-Anonymity. □

Note that this finding does not stand in conflict with the findings about attribute linkage, as those have been made under the original assumptions of the notions.

Since we just showed that Attribute Privacy weakly implies 2-Anonymity, it is relevant to point out that there is related work showing implications between t-closeness and

ε-Differential Privacy [23, 22, 31]. However the results of [22] do not fit neatly in the hierarchy we are building:

- For the implication of DP to t-closeness: k-Anonymizing the QIDs of a dataset *and* ensuring ε-Differential Privacy for the confidential attributes, achieves stochastic t closeness (their stochastic extension to t-closeness.)

  - "Attaining standard t-closeness from ε-differential privacy is unfeasible because, being ε-differential privacy a stochastic mechanism, it is not possible to meet the requirements that t-closeness puts on the empirical distribution of the confidential attributes"

- For the implication of t-closeness to DP: The authors use the assumptions made by t-closeness about the prior- and posterior knowledge of the adversary, but more importantly they show this only for the projection on the sensitive attributes because "t-closeness offers no additional protection to the quesi identifiers beyond what k-anonymity does. For example, we may learn that an individual is not in the data set if there is no equivalence class in the released t-close date whose quasi-identifier values are compatible with the individual's".

  - "Attaining exact ε-differential privacy from t-closeness is unfeasible because, for instance, t-closeness fails to provide any protection if we assume that all the records except one are known by the intruder."

This is not to say that these results are not meaningful in general, but rather that we can not combine them with what we found about the relations between the topological notions and the state-of-the-art.

Intuitively, as the notion of Attribute Privacy and Association Privacy prescribe strict deterministic structures over a given published relation, a mechanism achieving either one of them would not be ε-Differential Privacy, and likewise the output of an ε-Differential Privacy mechanism is not very likely to always output a relation that is attribute- or association-private. For completeness, we prove these intuitions for Attribute Privacy.

To attempt to compare Attribute Privacy and ε-Differential Privacy, we first need to formulate a semantic version of Attribute Privacy:

Definition 4.2.9 (Semantic Attribute Privacy)**.** A Mechanism $\mathcal{M}$ fulfils *semantic Attribute Privacy*, iff for all input relations R:

$$\mathcal{M}(R) = R'$$

R' fulfils Attribute Privacy.

Analyzing the relationship between Semantic Attribute Privacy and ε-Differential Privacy reveals, that the two notions are orthogonal to each other:

Proposition 4.2.10. *Semantic Attribute Privacy does not imply ε-Differential Privacy and ε-Differential Privacy does not imply Semantic Attribute Privacy.*

Proof. We prove this by counter-examples
  DP$\nRightarrow$ Attr Priv:
Consider the randomized response mechanism. For a given response with r option the returned response will be determined by a matrix of probabilities, with the real response

one one axis and the returned response on the other [23]. In the case where there are only two options (like in the relations handed by Attribute Privacy) this reduces to a statement controlled by a single probability p, where p is the probability that the response is the truth. The 2×2 matrix looks as follows

$$P = \begin{pmatrix} p & 1-p \\ 1-p & p \end{pmatrix}.$$

On the diagonal are the probabilities for truthful answers where the property was not observed and where the property was observed. The other two entries are for when the property was not observed, but it is reported that it was observed, and for when the property was observed but it is reported that it was not observed. Randomized response is a mechanism fulfilling DP [23]. Specifically for relations it is ε-Differential Privacy if

$$\exp(\varepsilon) \geq \frac{\max\{p, p-1\}}{\min\{p, p-1\}}.$$

Now lets say there is some relation given as a $m \times n$ table, which does not have Attribute Privacy. When applying randomized response with probability of returning the true response being p, there is a $p^{m \cdot n}$ chance that the returned relation will be the original relation. Since the original relation does not have Attribute Privacy in this case the output also does not have Attribute Privacy. This leads us to conclude that randomized response does not have semantic Attribute Privacy.

Thus we have found a mechanism that has ε-Differential Privacy, but not semantic Attribute Privacy.

Attr Priv $\not\Rightarrow$ DP:
Given a deterministic mechanism $\mathcal{M}$ that is ε-Differential Privacy will always be the constant mechanism [25, p. 4]. However consider the mechanism $\mathcal{M}$ which for a given input-relation removes the free faces from the attribute complex, and does so in a way that produces the minimal amount of changes. Should this algorithm use some randomness, via a random input r, it can be made deterministic by always providing a fixed $r^\star$. This deterministic mechanism $\mathcal{M}(\cdot, r^\star)$ has Semantic Attribute Privacy, as it always outputs relations with Attribute Privacy. $\mathcal{M}(\cdot, r)$ is not a constant mechanism: Consider one input relation T that does not have Attribute Privacy. The output of $\mathcal{M}(T, r^\star)$ will be some relation T' ($\neq T$) which has Attribute Privacy. Now consider some input relation $S \neq T'$ which has Attribute Privacy and no free faces. Since the mechanism minimizes the amount of changes needed to remove free faces, $\mathcal{M}(S, r^\star) = S$.

Since $\mathcal{M}$ is deterministic but not constant, $\mathcal{M}$ cannot have ε-Differential Privacy. $\square$

4.2.3. Attack Resilience Analysis

In this section we compare how the different notions withstand different known attacks. We follow the attack models introduced by [37] as these attack models have established themselves as a widely used tool to compare notions in SDC. We analyze the resilience of the topological notions against linkage and probabilistic attacks.

A linkage attack [37] is an attempt to re-identify individuals in an anonymized dataset by combining that data with background information. The linking may use quasi-identifiers, such as zip or postcode, gender, salary, and so on that are present in both sets to establish identifying connections. Within this group we find tree main subcategories: Table, Record and Attribute linkage.

4.2.3.1. Table Linkage

The presence or absence of a target's record in a database can reveal sensitive information. For example, the presence of an individual in a medical research dataset regarding diabetes patients, could be used by an U.S. employer (who would need to pay for medical expenses) to illegally discriminate against an individual [73]. Table linkage [73] occurs if attacker can infer presence or absence of target's record in released table thanks to the contrast with background information.

ε-Differential Privacy is resilient against table attacks. The advantage of an arbitrary attacker detecting the presence or absence of a target in the table is bounded by $\frac{e^\varepsilon - 1}{e^\varepsilon + 1}$ [47].

But k-Anonymity, ℓ-Diversity and t-Closeness do not protect against this kind of attack [87, 37].

Topological notions are established under the assumption in which membership is not sensitive. Therefore, no further investigation has yet been done about their resilience against table linkage. We investigate this matter and proof that both association and Attribute Privacy are vulnerable against table linkage.

Proposition 4.2.11. *Association Privacy and Attribute Privacy are vulnerable to table linkage.*

Proof. Assume target individual x^* has a global quasi-identifier $\text{GQID} \coloneqq \{a_1, \ldots a_i\}$. This is a combination of attributes that is unique to that individual, not only within one dataset, but within every possible dataset that contains those attributes.

$$\forall R \in X \times Y \quad \text{with } \text{GQID} \subseteq Y:$$
$$\{x \in X \mid Y_x \supseteq \text{GQID}\} \subseteq \{x^*\}.$$

Now consider a relational database which contains all of these attributes and their negations potentially among other attributes. That relation can contain the global identifier and still fulfil Attribute or Association Privacy. As soon as the global quasi identifier of that individual appears in that relation, an attacker knows they are a member of that relation.

We prove this via examples. Let $\text{GQID} \coloneqq \{a_1, a_2, a_3\}$. First, consider relation $R_{\text{GQID}}^{\text{attr}}$ as defined in Table 4.5. Its attribute complex is an empty octahedron, shown in Figure 4.1. It has no free faces, as the maximal faces are triangles, and all of the corresponding edges are edges of two triangles each. This means this relation has Attribute Privacy. At the same time the relation contains GQID, and thus is prone to table linkage.

Next, consider relation $R_{\text{GQID}}^{\text{assoc}}$ as defined in Table 4.6. Its association complex is two single vertices, shown in Figure 4.2. The association complex has no free faces. At the same time the relation contains GQID, and thus is prone to table linkage. $\qquad\square$

4.2.3.2. Record Linkage

The goal of an attacker in a record linkage attack is to determine which exact database record is the target. For this purpose, an attacker who knows that the target record is a member of the database uses the target's QID as background knowledge [37].

Knowing a target's entry in a database can be sensitive in itself, even if that database does not directly reveal more attributes, for example, when the order of the database itself is sensitive. But this can also be used as a step of a more involved attack, if the same database-identifier is used across multiple databases.

$R_{\text{GQID}}^{\text{attr}}$	a_1	$\neg a_1$	a_2	$\neg a_2$	a_3	$\neg a_3$
1	•		•		•	
2		•		•		•
3	•	•				•
4		•	•		•	
5	•			•	•	
6	•			•		•
7		•	•			•
8		•		•	•	

Table 4.5.: Relation with Attribute Privacy, containing GQID

Figure 4.1.: Attribute complex of $R_{\text{GQID}}^{\text{attr}}$

$R_{\text{GQID}}^{\text{assoc}}$	a_1	$\neg a_1$	a_2	$\neg a_2$	a_3	$\neg a_3$
1	•		•		•	
2		•		•		•

Table 4.6.: Relation with Association Privacy, containing GQID

Figure 4.2.: Association complex of $R_{\text{GQID}}^{\text{assoc}}$

k-Anonymity addresses this attack [37], by ensuring a minimum size for each group of records with the same QID.

Record linkage as defined by [37] cannot be applied to ε-Differential Privacy. This is because, ε-DP is a semantic notion, which means the mechanism is ε-DP and there is not one published ε-DP table where the attacker could identify the target [27]. Further ε-DP statistically bounds the influence a single individual can have on the output [27].

For the topological notations we obtain the following result:

Proposition 4.2.12. *Neither Association Privacy, nor Attribute Privacy hold against a record linkage attack.*

Proof. We prove this with the same counterexamples as before: Assume the QID of the target is QID $= \{a, b, c\}$, and the attacker knows that the target participated in the study, that produced the published table.

Let the published relation be $R_{\text{GQID}}^{\text{attr}}$ as defined in Table 4.5. It has Attribute Privacy, as we have previously shown. Knowing the QID, the adversary can identify the target as individual 1.

Let the published relation be $R_{\text{GQID}}^{\text{assoc}}$ as defined in Table 4.6. It has Association Privacy, as we have previously shown. Knowing the QID, the adversary can identify the target as individual 1. $\qquad\square$

4.2.3.3. Attribute Linkage

Attribute linkage occurs if the attacker can infer sensitive attributes of their target from published data, using their background knowledge on the target [37]. Note that topological notions do not assume a known quasi-identifier by which this happens, like k-Anonymity does. It is always applicable if the attacker can infer some previous unknown information

about their target given some some attributes of their target and the targets presence in the table

k-Anonymity does not protect against attribute linkage. This is because it can happen that the target is in an anonymity-group, but all members of that group have the same set of sensitive attributes. For example in a database where the quasi-identifier is $\langle \text{age}, \text{sex}, \text{zip-code} \rangle$ and all people that have the same age, sex and zip code as Alice, have debt, it can be deduced that Alice has debt. While this can be rectified by enforcing ℓ-Diversity [37], we still need to assume the set of QIDs which may lead to a successful attack is the background knowledge differs from our assumption

Attribute linkage is covered by ε-Differential Privacy when the data is uncorrelated. Since ε-Differential Privacy implies attribute-ε-Differential Privacy [51] and attribute-ε-Differential Privacy means that the significance of an attacker hypothesis inference trying to know if a certain attribute was or was not in a user record is statistically bounded [51]. When the data is correlated ε-Differential Privacy encounters certain limitations in this regard, we discuss this in section 4.3.

As for the topological notations:

Proposition 4.2.13. *Attribute Privacy does cover attribute linkage.*

Intuitively this is because Attacker cannot learn any new attributes about target, starting from a know set of attributes γ.

Proof. We proceed by reduction ad absurdum. Suppose there is a successful attribute linkage attack on a given relation R that satisfies Attribute Privacy. This means that there is at least one target for which the attack succeeds, which means:

1. The attacker has as background knowledge a set of (non-sensitive) attributes $\{y_1, \ldots, y_n\} =: \gamma$ of the target.

2. Is able to identify a group of individuals that have all of these attributes

$$\{x_1, \ldots, x_m\} =: \sigma \,,$$

3. and can infer sensitive attributes $\{y'_1, \ldots, y'_{m'}\} = \gamma' \setminus \gamma$ about the target from the attributes, that all members of the group share.

And since the attack is successful that means that $\gamma' \setminus \gamma \neq \emptyset$.

In other words, the attacker has applied $(\phi_R \circ \psi_R)$ to γ and received a γ' with $\gamma' \setminus \gamma \neq \emptyset$. This is in conflict with the assertion that the relation has Attribute Privacy. $\square$

Proposition 4.2.14. *Association Privacy does not protect against attribute linkage.*

Proof. We prove this by counterexample. Let $S \in X \times Y$ be a relation as shown in Table 4.7, Assume attacker knows Alice is in the relation (is in X)), and knows Alice has attribute a. It is clear that S has Association Privacy. We prove by exhaustion, that $\psi_S \circ \phi_S = \text{id}$:

S	a	b	c
1	•	•	
2			•

Table 4.7.: Relation S

$$
\begin{aligned}
((\psi_S \circ \phi_S)(\emptyset) \quad &= \psi_S(\{a,b,c\}) = \emptyset \\
((\psi_S \circ \phi_S)(\{1\}) \quad &= \psi_S(\{a,b\}) \quad = \{1\} \\
((\psi_S \circ \phi_S)(\{2\}) \quad &= \psi_S(\{c\}) \quad\; = \{2\} \\
((\psi_S \circ \phi_S)(\{1,2\}) &= \psi_S(\emptyset) \quad\;\; = \{1,2\}
\end{aligned}
$$

Despite S being association private, the attacker is able to infer, that Alice has attribute b. □

4.2.3.4. Attribute linkage with colluding adversaries

The attribute linkage attack model assumes, that information an attacker has, is only about the target themselves. If however the attacker has more information, for example about two people in the database because of two colluding adversaries, then a database which fulfills the notion of Attribute Privacy is vulnerable against attribute linkage.

Example 4.2.15. In this example, the attributes $\langle a, b, c \rangle$ represent the quasi-identifier, the attacker knows their target Bob has attributes a and b and tries to get sensitive attribute d. In addition to that they have background-knowledge. This comes in the form of relation BG but also in the knowledge, that both Bob and Alice participated in the study that produced the published table Attr.

Attr is defined in Table 4.8, and BG is defined in Table 4.9.

Attr	a	b	c	d
1	•	•	•	
2	•	•		•
3			•	•
4	•		•	
5	•			•
6		•	•	
7		•		•

Table 4.8.: Published relation

Figure 4.3.: Attribute complex of Attr

BG	a	b	c
ALICE	•	•	•
BOB	•	•	
CHARLIE			•
DOUGH		•	
EMILY	•		
FRANK	•		•
GLADYS		•	•

Table 4.9.: Background Knowledge

Relation Attr has Attribute Privacy.

Proof. In Figure 4.3 it can be seen, that the maximal faces of the attribute complex are

$$\{\{a,b,c\},\{a,b,d\},\{c,d\}\}.$$

With this we see that the free faces are $\{a,d\}, \{b,d\}, \{a,c\}$ and $\{b,c\}$. Each of those free faces there is an individual in ATTR, that produces only that face respectively (Individuals $5, 7, 4$ and 6). With Theorem 3.0.1, we have that ATTR has Attribute Privacy. □

Now using that $Y_{\text{Bob}} \subset \{a,b\}$, it can be determined that Bob is in group $\{1,2\}$. Using the background knowledge, the attacker determines that attribute c is not observable for

Bob (while it is observable for Alice). Since the attacker knows both participated in the survey, Alice's record must be 1 and Bob's record must be 2. It follows, that Bob does have sensitive attribute d.

4.2.3.5. Probabilistic Attack

A probabilistic attack would be successful if an attacker would change their prior probabilistic belief on sensitive information of the target posterior of seeing the published data [37]. In contrast to the previous attack models, the probabilistic attack model does not focus on the exact record or attribute that are leaked, but rather on the amount that a probabilistic belief of an attacker would change after seeing the published data. The goal for a notion under a probabilistic attack model is to bound this amount of change in beliefs to some small value [37]. A drastic change in the adversaries beliefs after seeing the data is considered a successful attack.

ε-Differential Privacy is resistant to probabilistic attacks. Per construction released data based on a database without targets entry is ε-indistinguishable to the released data based on database with targets entry. So an attacker cannot make any meaningful (probabilistic) deduction about the target.

A Database that has k-Anonymity and ℓ-Diversity can still be still vulnerable to this attack. Additional t-Closeness is needed to achieve resilience [37].

Proposition 4.2.16. *The topological notions are not resilient to probabilistic attacks.*

Attribute Privacy. Consider relation R3 as defined in Table 4.10. This relation has

R3	a	b	c	d
1	•	•	•	
⋮	⋮	⋮	⋮	
92	•	•	•	
93	•	•		•
94	•		•	
95		•		
96		•	•	
97	•			•
98		•		
99			•	
100				•

Table 4.10.: Relation R3

Attribute Privacy, but by knowing the target is in the database, the attacker knows the target has attributes a, b, and c with a probability of 92%. □

Association Privacy. Consider relation R4 as defined in Table 4.11. It has Association Privacy, but knowing the target is in the database, the attacker knows the target is associates with users 1, 2, and 3 with a probability of 92%. □

R4	a_1	...	a_{92}	a_{93}	a_{94}	a_{95}	a_{96}	a_{97}	a_{98}	a_{99}	a_{100}
1	•	...	•	•	•			•			
2	•	...	•	•		•	•		•		
3	•	...	•		•		•			•	
4				•				•			•

Table 4.11.: Relation R4

4.2.3.6. Comparing Results

In the above analysis, we have analyzed the resilience of the topological notions against different attack models, and have compared the results to the results achieved by [37]. We have seen that Attribute Privacy is only resistant against attribute linkage, and Association Privacy is not resistant against any of the models tested.

It is to note that the ascribed purpose of Association Privacy is to hide connections between people, not to protect individuals on their own, so it is not surprising that it fails against all of these attack models. However we have still included it in our analysis, as these models purpose is to compare a vast array of different notions. Further it is helpful to confirm the failure as opposed to solely argue that a notion was not designed with an attack model in mind, so it must fail against that model. Unfortunately, the work that introduces the topological notions [33] does not provide an attack model that we could have used for further comparison, and we were not able to find independent models that consider this use case.

We present an overview of above analysis and prior results in Table 4.12. In this table, a checkmark (✓) denotes if the presented notions withstands against the analyzed attack model, and a cross (✗) denotes if it does not.

	Table Linkage	Record Linkage	Attribute Linkage	Probabilistic Attack
Attribute Privacy	✗	✗	✓	✗
Association Privacy	✗	✗	✗	✗
k-Anonymity	✗	✓	✗	✗
ℓ-Diversity	✗	✓	✓	✗
t-Closeness	✗	✗	✓	✓
ε-Differential Privacy	✓	✓	✓	✓

Table 4.12.: Summary of which notions hold up against which attack model

4.3. Attribute Privacy plus Differential Privacy

Wang et al. [87] proposed the idea of combining topological notions with DP in order to solve some DP problems such as the vulnerability with respect to correlation-based attacks. They introduce the notion of "APL-free ε-Differential Privacy", where APL stands for Attribute Privacy linkage, and the authors do so too in reference to [33]. They argue, that correlations between the attributes allow an attacker to execute such APL attacks.

First, we notice, that "APL-free" is a relaxation of Attribute Privacy. We will formally prove this observation in subsection 4.3.1. Secondly we are not convinced, that [87]

show an issue that is present in correlated DP data. Nevertheless, correlations between attributes *can* lessen the guarantees of DP and Attribute Privacy could help with that. So in the following we

1. Show that APL-free is a relaxation of Attribute Privacy.

2. Show how [87] falls into inference privacy fallacy.

3. Analyze whether a combination of Attribute Privacy and DP can prevent an attribute-correlation-based attack, which would otherwise be possible on data that was only anonymized with DP.

4.3.1. APL-free is a relaxation of Attribute Privacy

The authors do not directly give a definition for APL-free, rather they give a definition for APL which they then use in their definition of APL-free ε-Differential Privacy:

Definition 4.3.1 (APL [87])**.** Given a dataset D, D has "Attribute Privacy leakage" (APL) if there is an itemset $Q \subset \mathcal{I}$ of the item universe $\mathcal{I}$ such that

$$\left| \left\{ S | S \in D^I \wedge Q \subset S \right\} \right| = 1 \, ,$$

where D^I is the set of itemsets in D. And by using the itemset Q the adversary can uniquely identify S in D^I and $S \setminus Q$ is the leaked information.

An adversary has the following information in advance:

1. The target's itemset S is in D.

2. The adversary knows a boundary itemset $Q \subset S$, where $Q \neq \emptyset$ and Q can uniquely identify S in D.

By accessing DP protected dataset D_p, the adversary will find the set of itemsets containing Q in D_p: $G = \left\{ S' | S' \in D_p^I \wedge Q \subset S' \right\}$. Then Q is an APL in D_p and the attack is successful if and only if $G = \{S\}$.

Definition 4.3.2 (APL-free ε-Differential Privacy [87])**.** A randomized algorithm $\mathcal{M}$ satisfies APL-Free ε-Differential Privacy, if $\mathcal{M}$ satisfies the following requirements:

1. For any $D_p \in \text{Range}(\mathcal{M})$, there is no APL in D_p.

2. For any two neighboring datasets D_1, D_2 and for any possible output $D_p \subseteq \text{Range}(\mathcal{M})$,
$$\frac{\Pr[\mathcal{M}(D_1) = D_p]}{\Pr[\mathcal{M}(D_2) = D_p]} \leq \exp \varepsilon \, .$$

We see the condition of $G \overset{!}{=} \{S\}$ to be arbitrary and unnecessary. To give an intuition, as to why APL-free is a relaxation of Attribute Privacy, consider the following: Let $G = \{S_1, S_2, \dots\}$ with $\bigcap_i S_i =: S^* \supsetneq Q$. According to the prior definition this would not be APL, since there exists no single S such that $Q = \{S\}$. However $S^* \setminus Q$ still leaks attribute information.

Proposition 4.3.3 (APL-Free is a relaxation of Attribute Privacy)**.** *Any given dataset D that has Attribute Privacy is also APL-Free, but a dataset D' that is APL-Free is not necessarily attribute private.*

Proof. To show this proposition, we will show that APL-free does not imply Attribute Privacy, and that Attribute Privacy does imply APL-free.

APL-Free $\not\Rightarrow$ Attribute Privacy: Let R be a relation, on which for any $Q \subset \mathcal{I}$

$$\left| \left\{ S | S \in D^I \wedge Q \subset S \right\} \right| \neq 1 \,,$$

however let there be one Q^* with $G^* := \left\{ S | S \in D^I \wedge Q^* \subset S \right\}$, and $S^* := \bigcap_{S_i \in G^*}$, where

$$|G^*| > 1 \wedge S^* \supsetneq Q^* \,.$$

Since there is no Q (including Q^*) for which $\left| \left\{ S | S \in D^I \wedge Q \subset S \right\} \right| = 1$ the relation is APL free. However the relation does not have Attribute Privacy: Denote x_i the individual from which S_i was produced. Since $S^* \neq Q^*$ and

$$\phi_D \circ \psi_D(Q^*) = \phi_D(\{x_1, \ldots, x_n\}) = S^*$$

we can conclude, that $\phi_D \circ \psi_D(Q^*) \neq \mathrm{id}$

Attribute Privacy $\Rightarrow$ APL-Free: Assume there exists a dataset D which has Attribute Privacy but is not APL-free (has APL). This means that there exists an itemset Q of attributes such that there is exactly one individual (x) that has all items of Q (as per APL definition). This individual has itemset S with $Q \subset S$, and since this is an APL $S \setminus Q \neq \emptyset$, so $Q \subsetneq S$. This means

$$\phi_D \circ \psi_D(Q) = \phi_D(\{x\}) = S \neq Q \Rightarrow \phi_D \circ \psi_D \neq \mathrm{id} \,.$$

Which stands in contradiction with the assumption that D had Attribute Privacy. $\qquad \square$

Since we see no reason to use a relaxation of Attribute Privacy, we suggest fully combining both notions:

Definition 4.3.4 (Attribute Private ε-Differential Privacy)**.** A randomized algorithm $\mathcal{M}$ satisfies *Attribute Private ε-Differential Privacy* (Attr-DP), if $\mathcal{M}$ satisfies the following requirements:

1. For any $D \in \mathrm{Range}(\mathcal{M})$, D fulfils Attribute Privacy, as defined in Definition 2.5.1.

2. For any two neighboring datasets D_1, D_2 and for any possible output $S \subseteq \mathrm{Range}(\mathcal{M})$,

$$\frac{\Pr[\mathcal{M}(D_1) \in S]}{\Pr[\mathcal{M}(D_2) \in S]} \leq \exp \varepsilon \,.$$

Recall Table 4.12. By construction, Attr-DP should be resistant against table linkage, record linkage and probabilistic attacks. However keep in mind, that DP is only vulnerable against attribute linkage, when attributes are correlated. This notion thus is only useful, if Attribute Privacy still holds even under correlations.

4.3.2. Inference privacy fallacy in APL

Now let us get into problems we see when it comes to how [87] addresses correlations. The problem of correlated data in differential privacy can be split up between correlations between individuals and correlations between attributes. APL is introduced as a potential attack vector that arises from attribute-correlated data.

Usually, when the correlation between attributes is discussed as a problem for DP, we talk about an adversary using prior knowledge about correlations between attributes, to lessen the privacy budget achieved [95]. As an example: Two attributes that are correlated 100% give the adversary two samples that have different noise applied, thus enabling differential analysis.

However the correlations used by the APL attack are not based on background knowledge of the adversary, but rather *inferred from the published data*. This leads into the *inference privacy fallacy* described by [52, p. 3], where "when the unwanted inference depends on generalizable (statistical) knowledge contained in the public dataset rather than on the specific datum that a target individual may or may not have contributed to the underlying confidential database". In the case given for APL the published data consists of rows of itemsets and occurrence counts for these itemsets. In this data a correlation is found between the given input itemset Q and some itemset $S \supset Q$ contained in the published database.

However keep in mind that this data is DP, so if the number of occurrences for the itemset S is low, the target has plausible deniability for any inference gained from $S \setminus S$. In the example given by [87], the itemset S occurs 23 times in the published dataset D_p which totals 40 occurrences. Additionally there is no other itemset $S' \supset Q$ that occurs in D_p. This is then used together with background knowledge about the targets participation in the survey and the itemset Q of the target to infer that the target has itemset S. This correlation between the Q and S is generalizable knowledge and not specific to the target. If the target had not participated in the survey the number of S occurrences in D_p would have been very similar the prior number.

4.3.3. Attribute Private DP vs. Correlation-Based Attacks

We just elaborated our issues with the concept with APL in DP data as it was presented by [87]. But correlated attributes in DP protected data can case issues, if the attacker knows about them as background knowledge. Here we will elaborate on the risks of correlated attributes and analyze whether requiring Attribute Privacy can help against some attack enabled by correlated attributes. As elaborated in [29] event-level privacy is concerned with hiding the presence or absence of a single event and user-level privacy the presence or absence of a single user. User-level privacy suffers under correlated users [95] and event level privacy suffers under correlated attributes [15].

Consider Table 4.13, which shows an example of correlated users. The data under consideration has the format shown in Table 4.13a. The users 1 and 2 can either be positively correlated meaning that with a high likelihood $x_1 \Leftrightarrow x_2$ or negatively correlated, where with a high likelihood $x_1 \Leftrightarrow \neg x_2$. Table 4.13b and Table 4.13c respectively show the probability distributions in both of those correlation cases.

When the data is published using a counting query of $f(x_1, x_2) = x_1 + x_2$ [95] prove, that a for a weak attacker the distinguishability of the probability distributions is outside the $[e^{-\varepsilon}, e^{\varepsilon}]$ bounds. They also show that in this particular case, the negative correlation does not show the same adverse effect. Note that this does not mean that negative correlation

	x
1	x_1
2	x_1

$\rangle$ Correlation

(a) Data format

	$x_1 = 0$	$x_1 = 1$
$x_2 = 0$	0.49	0.01
$x_2 = 1$	0.01	0.49

(b) Positively correlated

	$x_1 = 0$	$x_1 = 1$
$x_2 = 1$	0.01	0.49
$x_2 = 0$	0.49	0.01

(c) Negatively correlated

Table 4.13.: User-correlated data

can not have a negative effect similar to the one observed with positive correlation in this case.

Now consider the relation presented in Table 4.14, where attributes a and b are correlated. This poses a corresponding problem for event-level ε-Differential Privacy. If Attribute Privacy manages to stay unaffected by these correlations, then a combination of these notions, like in Definition 4.3.4, will be able to protect against such attribute-correlation attacks. Note that the correlation problem in DP only occurs under a weak adversary, that does not have background knowledge of other individuals in the relation. This means that in a combined notion, the protections of Attribute Privacy would only become relevant, when facing a weak adversary.

Correlation

	a $\frown$ b $\cdots$
1	a_1 b_2
2	a_2 b_2
$\vdots$	

Table 4.14.: Attribute-correlated data

We analyzed if Attribute Privacy manages to stay unaffected by attribute correlations, under the following case:

- Some Relation R_{orig}, which does not fulfil Attribute Privacy, has some mechanism applied resulting in modified relation R_{pub} that does, and this modified relation then gets published.

- The adversary has background knowledge of some strong correlation between two attributes, which leads them to find a Relation $R_{\mathrm{pub}}^{\mathrm{corr}}$ which is closer to the original relation.

We then analyzed if $R_{\mathrm{pub}}^{\mathrm{corr}}$ still has Attribute Privacy. We found that Attribute Privacy is vulnerable with some correlations, but it is also able still hold up under at least one very specific correlation.

Proposition 4.3.5. *Given an original relation without Attribute Privacy, the corresponding published relation R, which fulfils Attribute Privacy, with at least two properties a and b, and a correlation $a \Rightarrow b$. An adversary which knows the strong positive correlation $a \Rightarrow b$ can construct a relation R^+. Aside from the information already contained in the background knowledge, R^+ still has Attribute Privacy.*

Proof. Recall that for Attribute Privacy to hold for a relation R the following has to be true

$$\phi_R \circ \psi_R \text{ is identity operator on } \mathfrak{F}(\Phi_R) \cup \{\emptyset\}$$

So for a given set of attributes σ_1, ψ_R will map it to individuals γ, and ϕ_R will map the individuals to their shared attributes σ_2. And only if $\sigma_2 = \sigma_1$ does R fulfil Attribute Privacy. Let

- σ_1 a set of attributes, $a \in \sigma_1$

- $\gamma := \psi_R(\sigma_1); \quad \sigma_2 := \phi_R(\gamma)$

- $\gamma^+ := \psi_{R^+}(\sigma_1); \quad \sigma_2^+ := \phi_{R^+}(\gamma^+)$

- γ^* all individuals affected by (and whose attributes changed because of) the correlation (that previously had a but not b)
    - $\phi_R(\gamma^*) = \sigma_R^*; \quad a \in \sigma_R^* \land b \notin \sigma_R^*$
    - $\phi_{R^+}(\gamma^*) = \sigma_{R^+}^*; \quad a, b \in \sigma_{R^+}^*$

Now we analyze different possible values for γ_+:

- If $\gamma = \gamma+$ "Set of individuals unchanged":
    - If $\gamma^+ \subseteq \gamma^*$ then $\sigma_2^+ = \sigma_2 \cup \{b\}$ (All individuals gained b).
    - Else: $\exists j \in \gamma^+ : j \notin \gamma^*$
        * This means j either already had b in R or does not have a or b in R^+.
        * In both cases $\sigma_2^+ = \sigma_2$.

- Else if $\exists i \in \gamma^+ : i \notin \gamma$ "gain individual":
    - Only possible if $b \in \sigma_1$ ($\Rightarrow b \in \sigma_2 \land b \in \sigma_2^+$)
    - then $\gamma^+ \setminus \gamma = \gamma^* \Rightarrow \sigma_2^+ = \sigma_2 \bigcap_{y \in \gamma^*} X_y$ (Set of shared attributes can only shrink with more individuals)

- Else ($\exists i \in \gamma : i \notin \gamma^+$) "loose individual"
    - Not possible with positive correlation

So when $\sigma_1 = \phi_R(\gamma^*)$ ($b \notin \sigma_1$), it can be the case that $\psi_R(\sigma_1) = \gamma^*$ and then $\phi_{R^+} \circ \psi_{R^+}(\sigma_1) = \sigma_1 \cup \{b\}$. And in all other cases $\phi_{R^+} \circ \psi_{R^+}(\sigma_1) = \sigma_1$. Note that for a different definition of σ_1, where $a \notin \sigma_1$, the correlation would affect none of the individuals in γ or γ^+ and thus $\gamma = \gamma^+$ and $\sigma_2 = \sigma_2^+$ would hold.

So while Attribute Privacy is not strictly held, the only attribute information that R^+ leaks is $a \Rightarrow b$, which was already contained in the background information. $\square$

Proposition 4.3.6. *Given an original relation without Attribute Privacy, the corresponding published relation R, which fulfils Attribute Privacy, with at least two properties a and b, and a correlation $a \Rightarrow \neg b$. An adversary which knows the strong negative correlation $a \Rightarrow \neg b$ can construct a relation R^-. There are cases in which R^- does not retain Attribute Privacy.*

Example 4.3.7. Consider relation R and R^- as depicted in Table 4.15. Let variables be as above, additionally

- $\gamma^- := \psi_{R^-}(\sigma_1); \quad \sigma_2^- := \phi_R(\gamma^-)$

R	a	b	c
1	•	•	
2		•	•
3	•		•

R^-	a	b	c
1	•		
2		•	•
3	•		•

(a) Published Relation   (b) Inferred Relation

Table 4.15.: Relations showcasing problem with negative correlations in Attribute Privacy

- $\gamma^\star$ all individuals affected by the correlation (that previously had a and b, but now only a)
  - $\phi_R(\gamma^\star) = \sigma_R^\star; \quad a, b \in \sigma_R^\star$
  - $\phi_{R^-}(\gamma^\star) = \sigma_{R^-}^\star; \quad a \in \sigma_{R^-}^\star \wedge b \notin \sigma_{R^-}^\star.$

So in this example $\gamma^\star = \{1\}$. Consider the "loose individual" case, so $\exists i \in \gamma : i \notin \gamma^-$. This can (only) happen when $b \in \sigma_1$ and $i \in \gamma^\star$. In this example $\sigma_1 = \{b\}$.

With $b \in \sigma_1$ it holds that $\gamma \setminus \gamma^- = \gamma^\star$. In this example $\gamma = \{1, 2\}$ and $\gamma^- = \{2\}$. Note that the set of individuals unaffected by the correlation $\gamma \setminus \gamma^\star$ is equal to γ^-. Now consider the case in which additionally all individuals in γ^- share an attribute x not shared by the individuals affected by the correlation. In this example, individuals $\gamma^-(= \{2\})$ share attribute c, while individuals $\gamma^\star(= \{1\})$ do not: $x = c$.

Then $x \in \sigma_2^- \wedge x \notin \sigma_2$. In this example $\sigma_2^- = \{b, c\}$ and $\sigma_2 = \{b\}$. Thus Attribute Privacy no longer holds for R^-.

So, since there are correlations, for which Attribute Privacy fails, as evidenced by the preceding example, we will not investigate the combination of Attribute Privacy and DP further. We would however like to point out, that in the example of [95] the positive correlation was problematic. So for specific applications in which attributes are correlated, it might make sense to analyze, if Attribute Privacy is unaffected by the kind of correlation and if that is the case, use Attribute Private ε-Differential Privacy.

5. Privacy for Graph Data

Typically, in SDC, our focus is on the privacy of individual subjects, specifically their presence and attributes. This is the context in which k-Anonymity and DP operate, and where we demonstrated that Association Privacy does not hold up.

However, in many real cases, such us social networks analysis, the private information we aim to protect may concern more than one individual, for instance, the friendship relationship between two users. This scenario is modeled through graph data, where typically a node represents an individual and edges represent relationships or shared attributes between those individuals. In many cases the protection goal over graph data is to prevent re-identification attacks, where an adversary can identify a target in the published graph. There are also scenarios in which the identities of the vertices, and even their attributes are not sensitive themselves, but the edges between target individuals are. One such scenario is a transaction network, in which an edge denotes a financial transaction between two individuals and is considered sensitive.

As mentioned before (section 2.5), the protection goal of Association Privacy is to protect relationships between people. Therefore, the goal of Association Privacy appears to align with the protection goals usually seen in SDC notions for graph data. Especially in scenarios such as the above mentioned transaction network. This motivates us to analyze its protection and potential usability in the graph protection domain.

In this chapter we take a closer look on Association Privacy in the context of graph data. We answer two questions: Is a relation that is protected by Association Privacy vulnerable against a state-of-the-art graph privacy adversary? Could Association Privacy be used to protect a graph?

As a prerequisite, in section 5.1, we list an existing classification of privacy breaches on graph data, as well as existing classification of graph data background knowledge. Further, we establish a taxonomy of existing syntactic notions in this field. To the best of our knowledge we are the first ones providing a summary of common syntactic graph privacy notions, with a classification in protection goals and assumed background knowledge. For completeness, we briefly summarize an existing rigorous taxonomy and analysis of semantic graph privacy notions [60] in section 5.2.

Then, in section 5.3 we answer the first question, by analyzing whether a graph, based on an association private relation, is vulnerable against a SOTA adversary. To do this we introduce a canonical method to create a graph from a relation. We then proceed to perform graph-privacy attacks on those graphs, showing, that Association Privacy does not, in fact, protect relations against the surveyed kinds of attackers. Using our taxonomy, we identity the SOTA syntactic SDC notion with matching protection goal: k-security [18]. We show that Association Privacy offers weaker protection, than k-security.

Finally, in section 5.4 we negatively answer the question, whether Association Privacy could be used to protect data that is initially given as graph data.

5.1. State-of-the-Art in Syntactic Graph Data Privacy

Multiple surveys have been released about private graph data [53, 1, 93]. Unfortunately these focus on "techniques" or "methods" to achieve some form of graph data privacy, and do not compare notions. Correspondingly they do not create a taxonomy of notions. Some of these "techniques" are the mechanisms of existing notions, while other "techniques" lack a corresponding formal notion altogether. Despite this, the surveys are still helpful to get an overview of relatively current notions and their challenges.

We identify a collection of notions, whose mechanisms have been discussed by all three of the surveys. We briefly summarize each of those notions in subsection 5.1.3. In Appendix A, we summarize and comment on the "method"-classifications that the surveys *did* provide, which lead us the the selection of notions we identified.

We aim to create a rough taxonomy for these notions, based on protection goals and assumed attacker background knowledge. For this purpose we gather existing classifications of privacy breaches and types of background information for graph data from existing works.

A common [53, 1, 62] classification of privacy breaches in graph data, and their respective protection goals is as follows:

1. Identity disclosure, where the adversary's goal is to identify the identity of the individual associated with a node

2. Link disclosure, where the sensitive relationships between individuals are disclosed, and

3. Content disclosure, where the privacy of additional data associated with a node or edge is breached. This kind of privacy breach is called "content disclosure" by [1, 62], and "attribute disclosure" by [53].

We note, that these categories are not formally defined by these works themselves. Link disclosure is most similar to the protection goal of Association Privacy. For this reason, we give a formal definition of link disclosure in subsection 5.3.1. A formal definition of the other types of disclosure is outside of the scope of this thesis.

Additionally we note, that content disclosure only applies over *rich graphs.* Rich graphs are graphs in which edges or vertices are annotated with additional attributes in their own right [93]. These kinds of graphs are often the data format of social networks, in which the nodes contain sensitive personal information, and the edges contain sensitive information about the kind of relationship the two individuals share [93]. Wu et al. note that for rich graphs, "it is imperative to study the impact on privacy disclosures when adversaries combine attributes and structural information together in their attacks" [93].

We do not believe that Association Privacy would be applicable to rich graphs, since it protects sensitive relationships, that are implied by the attributes of the individuals. Having attributes (in the form of vertex labels), that do not necessarily match the associations (which are represented as edges) would make it unclear what exact we would be trying to protect with Association Privacy.

5.1.1. Graph Data Background Knowledge

In the following we briefly summarize existing classifications of attacker background knowledge for graph data. We use this as a base to differentiate between existing graph

privacy notions in subsection 5.1.3. We also remark on the equivalent graph background knowledge, assumed by Association Privacy.

The surveys [1, 53] point out, that in contrast to DP-based notions, the non-DP notions and mechanisms surveyed rely on the analyst knowing the adversary's background knowledge. And [93] notes, that modelling all types of background data for social network data is is challenging, and that this makes it harder to design anonymization techniques for such data.

While Association Privacy does not assume a QID like k-Anonymity does, it does assume that the adversary knows a set of individuals, that all share the same association. In a graph setting this would be a subgraph of one vertex and its neighbors. This is only one of many possible types of background knowledge in a graph setting.

In the following we will list the different kinds of background knowledge that have been listed by [93]:

1. Attributes of vertices

2. Specific link relationships between target individuals

3. Vertex degrees

4. Neighborhoods structural information of target individuals

5. (Embedded) Subgraphs

6. Graph metrics (betweenness, closeness, centrality)

7. Topological similarity/ distance

8. Auxiliary information

   - This includes an auxiliary graph whose members overlap with the anonymized target graph and a set of probability distributions defined on attributes of nodes and edges.

The attackers discussed by [93] gain their background knowledge, either actively, by creating new user accounts, or passively by finding unique sub-graphs in the network pre-publication, in order to identify individuals post-publication. The survey [93] notes that a randomly generated subgraph formed by $\mathcal{O}\left(\sqrt{\log n}\right)$ nodes can compromise the privacy of an arbitrary target with high probability and that most nodes in real social network data already belong to a small uniquely identifiable subgraph.

5.1.2. Utility Classifications

Survey [93] notes that it is hard to quantify information loss for graph data. They classify utility in three types: Graph topological Properties (such as degree sequences, shortest connecting paths and clustering coefficients), graph spectral properties (with the spectrum being the set of eigenvalues of the graph's adjacency matrix, which has close relations with many graph characteristics) and aggregate network queries. Additionally they mention that the number of modified edges can also be used to quantify loss.

5.1.3. Existing Notions

In the following we look into existing graph privacy notions, whose mechanisms have been mentioned (but not defined) in the previously evaluated surveys. The surveys also did not provide a taxonomy for these notions. To provide some structure, in Figure 5.1 we put the notions we considered in relation with each other by protection goal and assumed background knowledge.
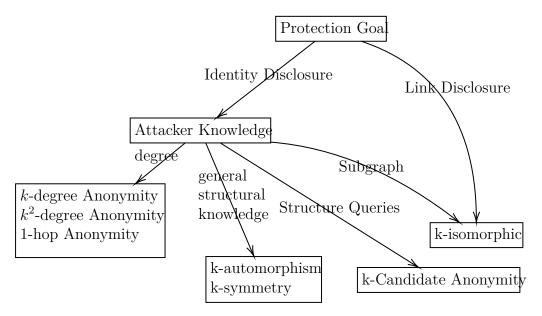

Figure 5.1.: Categorization of k-Anonymity based notions

5.1.3.1. k-Degree Anonymity

Introduced in [62]. Focuses on *identity disclosure.*

They denote $\mathbf{d}_G$ the *degree sequence* of G, a sorted vector of size equal $n = |V|$ such that $\mathbf{d}_G(v_i) = \mathbf{d}_G(i)$ is the degree of node v_i of G.

Definition 5.1.1 (k-degree Anonymity)**.** A vector of integers $\mathbf{v}$ is k-anonymous if every distinct value in $\mathbf{v}$ appears at least k times.

And a graph $G = (V, E)$ is *k-degree anonymous* if the degree sequence of G, $\mathbf{d}_G$ is k-anonymous

Adversaries have an a priori knowledge of the degree of certain nodes. Their anonymization problem consists of achieving k-degree anonymity by adding a minimal amount of edges.

According to [1] it "outperforms the random-based approach by achieving a greater degree of anonymity and less pertubation on graphs". Survey [93] notes about this notion, that the degree sequences of real-world graphs are highly skewed, and it is usually easy for adversaries to collect the degree information of a target individual.

We saw this notion used in multiple recent papers: "K-Degree Anonymity on Directed Networks" [16] adapts this notion for directed networks. "Personalized Privacy Preserving Method for Social Networks Graph K-Anonymization" [70] focuses on identity disclosure, and contributes a framework in which users are allowed to set their own privacy level requirements.

"Plausible Heterogeneous Graph K-Anonymization for Social Networks" [56] addresses heterogeneous graphs, so graphs with different types of edges and nodes. Their presented framework uses adversarial learning to address the attack vector of training a predictive model with the published graph to infer sensitive relationships. A big part of that framework is to enforce k-degree anonymity.

5.1.3.2. k^2-Degree Anonymity

Introduced in [80]. The stated goal is to prevent vertex re-identification, which means it falls under *identify disclosure.*

Their notion is specifically designed to prevent *friendship-attacks*, where the adversary knows the degree of two vertices that share an edge and tries to identify the first of these vertices. They refer to [62] (*k*-degree Anonymity) and related papers. In this context they state that while the other papers aim to prevent attacks based on vertex degrees their objective of their paper is to prevent friendship attacks based on vertex degree. A direct comparison with the attacker of [62] might be difficult because it generally assumes knowledge of the degree of "certain nodes", which could include a pair of connected nodes, however the additional knowledge that two of these nodes share an edge might be crucial.

Definition 5.1.2 (k^2-Degree Anonymity)**.** A graph $\bar{G}$ is k^2-degree anonymous if for every vertex with an incident edge of degree pair (d_1, d_2) there exist at least $k - 1$ other vertices such that each of the $k - 1$ vertices also has an incident edge of the same degree pair.

Tai et al. propose, that k^2 degree anonymity implies *k*-degree anonymity.

5.1.3.3. k-Degree Subset Anonymity

Introduced in [19], is a generalization of the degree anonymization problem proposed by [62]. The idea behind this notion is that in some cases the attacker is only interested in some subset vertices, and thus it is not necessary to anonymize the entire graph. It focuses on subset anonymity, where only a subset of the vertices need to be protected.

Definition 5.1.3 (k-Degree-Subset-Anonymity)**.** GIven an input Graph $G = (V, E)$ and an anonymizing subset $X \subseteq V$, produce an output graph $G' = (V, E \cup E')$ such that X is k-degree-anonymous and $|E'|$ is minimized.

5.1.3.4. k-Neighborhood

Introduced in [98]. The paper calls its notion *k*-anonymous, to avoid confusion, we rename this notion to $k - 1\text{-}neightborhood\text{-}anonymous.$

Survey [93] summarizes this notion as follows: "At least $k - 1$ other vertices in the graph such that the k induced subgraphs are isomorphic".

The *neighborhood* of a vertex u is the induced subgraph of the neighbors of u. This means the subgraph contains those vertices and all edges them in the original graph.

The privacy of a vertex u of the original graph G is considered leaked if an adversary can identify a vertex u' in the published graph G' that connects to other vertices in G' in a way that is similar to the way u connects to other vertices in G.

This means the goal of this notion is to prevent any vertex $u \in V(G)$ to be identified in G' with high confidence ($> 1/k$). This means this notion also addresses the *identity disclosure* problem.

Definition 5.1.4 (k-1-neighborhood-anonymous)**.** Given a graph G, for a vertex $u \in V(G)$, u is k-1-*neighborhood-anonymous in* G' if there are at least $(k-1)$ other vertices $v_1, \ldots, v_{k-1} \in V(G)$ such that

$$\text{Neighbor}_{G'}(\mathcal{A}(u)), \text{Neighbor}_{G'}(\mathcal{A}(v_1)), \ldots, \text{Neighbor}_{G'}(\mathcal{A}(v_{k-1}))$$

are isomorphic. G' is k-1-neighborhood anonymous, if every vertex in G is k-1-neighborhood-anonymous in G'

The paper [98] models social network as vertex-labelled-graph in which the labels form a hierarchy. The Assumed background knowledge is the neighborhood of some target individuals (). This these vertices are called the *d*-neighbors and are the vertices within distance *d* to the target vertex in the network. A large *d* is impractical for adversary, because social networks have small diameter, so a $d > 1$ adversary has to collect information about many vertices. The paper [98] discusses only $d = 1$, leaves $d > 1$ for future work.

5.1.3.5. 1-**hop Anonymity**

Introduced in [84], 1-hop Anonymity is supposed to protect against a type of degree based attack attempting re-identification. This means we can classify its protection goal as *identity disclosure*. Here, the adversary is assumed to have prior knowledge about the degrees of nodes within a given radius of the target node (denoted by the number of hops from the target node).

Definition 5.1.5 (k-Anonymous against i-hop degree-based attacks)**.** The *i-hop fingerprint* of a vertex $v \in V$ denoted f_i is the sequence of degree sets

$$d(v), \{d(u)|u \in N(v,1)\}, \ldots \{d(u)|u \in N(v,i)\}$$

And a graph $G(V, E)$ is k-*Anonymous against i-hop degree-based attacks*, if for each node $v \in V$ there exists at least $k-1$ other nodes $u_1, \ldots, u_{k-1} \in V$ such that

$$f_i(v) = f_i(u_j) \quad \forall j \in [1, k-1].$$

5.1.3.6. k-**Candidate Anonymity (vertex)**

Introduced in [45], k-Candidate Anonymity addresses *structure queries*. It ensures that for a given structure query that matches a candidate, at least $k-1$ other candidates match that same query.

Hay et al. note a *candidate set* of an individual $x \in V$ as $\text{cand}(x)$ and define it to consist of the nodes of the anonymized Graph G_a that could feasibly correspond to x. Similarly, if the adversary uses a query $Q(x)$ as background information the corresponding candidate set is called $cand_Q(x)$. They define a generalization of a graph as a Graph consisting of supernodes, superedges and an accompanying edge function $d(\cdot, \cdot)$. The supernodes $\mathcal{V}$ are a partition of nodes of the original vertex set V and the value of d corresponds to the number of edges in the original graph this superedge contains.

As a guarantee against re-identification they require each partition to have at least size k. They state that this ensures $cand_Q(x) \geq k$ for any adversary Q. As the notion guarantees against re-identification, it addresses the protection goal of *identity disclosure*. The authors assumption throughout the work is that external information sources are accurate, but not

necessarily complete: They "distinguish between a closed-world adversary, in which absent facts are false, and an open-world adversary in which absent facts are simply unknown". They define three variants of adversary background knowledge:

- Structural queries modeling an adversary with closed-world information about node degree

- Class of structural queries modeling weaker adversary, who explores the graph edge-by-edge with open world information

- Knowledge provided by hub-connections (hub node is a node with high degree and betweenness centrality)

They call their kind of attack "structral reindentification".

5.1.3.7. k-**Automorphism**

According to [93] k-Automorphism was developed to mitigate subgraph-based privacy attack. But has the disadvantage, that in order to make a graph symmetric for anonymization, adding new vertices and modifying the edges may be required, which will reduce utility.

From the at the initial paper [99] we find, that k-Automorphism is focussed on the *identity disclosure* problem, as opposed to link exposure. They state that with disclosed identities, some sensitive link information can be derived as well but acknowledge that the link-disclosure problem is orthogonal to identity-disclosure discussed in the paper.

The definition given for k-Automorphism is as follows:

Definition 5.1.6 (k-automorphic Network)**.** An automorphism of a graph is an automorphic function f of the vertex set V such that for any edge $e = (u, v)$, $f(e) = (f(u), f(v))$ is also in G.

A network G, is called a *k-automorphic network* if there exist $k - 1$ automorphic functions $F_a(a = 1, \ldots, k - 1)$ in G, and for each vertex v in G,

$$(F_{a_1}(v) \neq F_{a_2}(v)(1 \leq a_1 \neq a_2 \leq k - 1).$$

The paper [99] points out, that without this notion any topological structure of the network can be used to identify target individuals in the released network. This notion specifically targets structural attacks. For this reason we classify its background knowledge as *general structural knowledge*. We additionally note, that this notion allows for dynamic releases.

5.1.3.8. k-**symmetry**

Introduced by [92], k-symmetry targets *identity disclosure* under structural knowledge attacks.

The general idea of this notion is to modify the graph, in such a way, that for each vertex v there are at least $k - 1$ other vertices which serve as the image of v under some automorphism. The adversary is assumed to have some (generalized) structural knowledge about their target $v \in V$.

The automorphism equivalence introduces a vertex partition of vertices u, v for which there exist an automorphism $g \in Auth(G)$ such that $u^g = v$). The paper [92] calls this vertex partition the *automorphism partition* of G and denote it by $Orb(G)$.

Definition 5.1.7 (*k*-Symmetry Anonymity)**.** Given a graph *G* and an integer *k*, if

$$\forall \Delta \in Orb(G), |\Delta| \geq k \,,$$

then G is satisfies the requirement of *k-symmetry anonymity.*

The paper [92] states "*k*-symmetry anonymity is a generalization of any other *k*-anonymities of graphs based on different structural constraints on vertices. In other words, if a graph is *k*-symmetric, it also satisfies any other *k*-anonymity requirements defined in terms of other structural constraints on vertices, such as degree, neighborhoods and so on". The authors criticise [62, 98] and [45] for assuming some *specific* structural knowledge and point to the necessity for a *k*-anonymity model independent of structural knowledge used. They also take note of [99] (*k*-Automorphism) and write that whether *k*-automorphism is equivalent to *k*-symmetry still needs rigorous proof.

5.1.3.9. *k*-Isomorphic graph

Introduced in [18], *k*-Isomorphism is an expansion of *k*-candidate anonymity [93].

The notion, named *k-security*, models the adversary's background knowledge as a subgraph of *G* up to the entire given graph. One vertex in this neighborhood attack graph is always marked as the vertex under attack. This pair (G_a, v) is called the *NAG* (Neighborhood Attack Graph), concerning a target individual A, where *v* is the vertex belonging to A.

The notions goal is to publish a (labelled) graph in such a way that given some NAGs *vertex and link information* can not inferred with a higher probability than a set threshold:

Definition 5.1.8 (*k*-security)**.** Given a Graph $G = (V, E)$ with node-labels, given as $I(v)$, where every node represents a unique individual. The anonymized Graph G_k of *G satisfies k-security with respect to G*, if for any two individuals A and B with corresponding NAGs G_A and G_B (known to the adversary) the following conditions hold

- (NodeInfo Security) the adversary cannot determine from G_k and G_A that A is linked to $I(v)$ for any vertex *v* with a probability higher than $\dfrac{1}{k}$ (Analogue for B)

- (LinkInfo Security) the adversary cannot determine from G_k, G_A and G_B that A and B are linked by a path of a certain length with a probability of more than $\dfrac{1}{k}$

The title-giving *k*-Isomorphism is part of the solution given to the notion's problem statement. It consists of given a Graph *G*, reaching a *k*-secure graph with respect to *G* while minimizing a cost metric given in the paper.

Definition 5.1.9 (k-Isomorphism)**.** A graph *G* is *k*-isomorphic if *G* consists of *k* disjoint subgraphs $g_1, \ldots, g_k$ where g_i and g_j are isomorphic for $i \neq j$.

Given a Graph *G*, the solution to the problem statement is to derive a graph G_k with the same vertex-set such that G_k is *k*-isomorphic. The paper also discusses dynamic releases.

5.1.3.10. k-Obfuscation

Introduced by [11], k-Obfuscation has the goal to anonymize the graph G so that the identity of its vertices is obfuscated, which means it has the goal of protecting against *identity disclosure*. The notion quantifies the privacy-level via entropy.

The adversary is assumed to posses knowledge about some vertex property P of their target vertex. Examples given for this property include the degree of the vertex, of the vertex and of its neighbours, the neighborhood subgraph induced by the target vertex and its neighbors.

The notion works over uncertain graphs. An uncertain graph is defined in relation to some original graph $G = (V, E)$: The uncertain graph on the vertices of G is a pair $\tilde{G} = (V, p)$, where $p : V_2 \to [0, 1]$ is a function that assigns probabilities to unordered pairs of vertices (V_2). This means every possible edge given the vertex set V has some assigned probability of existing.

Definition 5.1.10 ((k, ε)-Obfuscation)**.** Let P be a vertex property, $k \geq 1$ a desired level of obfuscation and $\varepsilon \geq 0$ a tolerance parameter. The uncertain graph $\tilde{G}$ is said to k-*obfuscate* a given vertex $v \in G$ wrt. p if the entropy of the distribution $Y_{P(v)}$ over the vertices of $\tilde{G}$ is $\geq \log_2(k)$:

$$H(Y_{P(v)}) \geq \log_2(k).$$

The uncertain graph $\tilde{G}$ is a (k, ε)-*obfuscation wrt. property P* if it k-obfuscates at least $(1 - \varepsilon)|V|$ vertices wrt. P.

The tolerance parameter ε is there to compensate for outliers (such as celebrities in social networks). The goal for anonymization is to lower the bound of entropy of the distribution it induces over the obfuscated graph vertices.

This work is improved upon by [74]. They propose a general model called *uncertain adjacency matrix* (UAM) which captures (k, ε)-obfuscation. Given the original graph G_0 the UAM $\mathcal{A}$ of $\mathcal{G}$ must satisfy

1. $\mathcal{A}_{ij} = \mathcal{A}_{ji}$ (Symmetry)

2. $\mathcal{A}_{ij} \in [0, 1]$ and $\mathcal{A}_{ii} = 0$ (no multi-edges or self-loops)

3. $\sum_{j=1}^{n} \mathcal{A}_{ij} = d_i(G_0)$ $i = 1 \ldots n$ (expected degrees of all nodes must be unchanged; preserving the degree sequence)

Given this context, (k, ε)-obfuscation can be seen as an approach to construct an UAM while having the additional restriction over the entropy of the vertex property distribution as described above. Nguyen, Imine, and Rusinowitch state in [74], that in unlabeled graphs (where node identifiers are numbered in an arbitrary manner) an attacker aims at re-identifying nodes based on their structural information and that for this line of graphs *node privacy implies link privacy*. Additionally they state, that k-anonymity has the same semantics as k-obfuscation with the corresponding minimum entropy of $\log_2 k$.

5.2. State-of-the-Art in Semantic Graph Data Privacy

A recent survey "Private Graph Data Release"[60] introduces a taxonomy for semantic graph data privacy notions. In the following, we briefly summarize their findings, as

together with the previous section, this complements our understanding of the graph data SOTA to encompass both the syntactic and semantic notions. We highlight, that in the graph data context, there exist further semantic privacy notions that are not based on DP.

The tree-based taxonomy divides research on private graph data release on the first level in graph release or statistic/ query release. This means it makes the distinction whether the full graph will be released and thus has to be anonymized, or just some aspects derived from it. The former also includes synthetic graph releases.

The next level divides both branches each in provable and non-provable methods: "Provable privacy mechanisms offer mathematical guarantees about the privacy protection or the utility that they can provide. Non-provable privacy mechanisms do not provide such strong theoretical guarantees and thus are more empirical in nature". Here *k*-Anonymity-like approaches fall into the non-provable category, while DP-methods fall into the provable one. Broadly speaking, the survey's classification of provable methods aligns with what we would call semantic notions, while the non-provable category not only contains syntactic notions, but also mechanisms without any formal notion attached. They further criticize, that the "non-povable methods" make strong assumptions about the background knowledge of adversaries. They argue, that this is difficult to anticipate, and thus those methods do not provide strong privacy guarantees. Given this context, we had high hopes for the privacy guarantees of Association Privacy compared to syntactic notions, because it allows background knowledge of multiple individuals that are associated with each other. However, as we will show later, Association Privacy has weaker privacy guarantees than comparable syntactic notions.

The survey [60] is putting a heavy focus on these provable mechanisms, which mostly are extensions on differential privacy for graph data. Their reasoning for this is "DP's widespread use and acceptance in the research community". Like [53, 1, 93] this survey considers algorithms, however unlike the prior, it does give various formal definitions for the notions surveyed.

They discuss the advantages of statistic or query release of graph data over releasing the full graph:

- The information about the graph provided to the analyst is explicitly defined.

- In general, less noise needed because only specified queries need to be protected. With synthetic graphs, the class of all possible queries is much greater.

Whereas the advantage of (synthetic) graph release they point out is that (synthetic) graph releases are independent of graph queries, and can be used to answer subsequent questions with no or low risk of privacy leakage. On generative synthetic DP graphs they note "one way to synthesize a provably private graph is to ensure the parameters of such generative models are estimated in a way that is differentially private". This means, that like in the relational data case, for the equivalent of microdata releases, semantic notions likely come with high utility penalties, compared to syntactic notions.

Of the DP notions discussed, we took note of edge-DP, since it is supposed to protect the relationship between two entities, which is close in goals with Association Privacy. In contrast, node-DP protects the existence of an entity and its relationships with others.

Definition 5.2.1 (Edge neighboring [60])**.** Given a graph $G = (V, E)$, a graph $G' = (V', E')$ is an *edge neighboring graph* of G, if it differs from G by exactly one edge, i.e., $|V \oplus V'| + |E \oplus E'| = 1$, where $x \oplus y$ is the symmetric difference between sets x and y,

$x \oplus y = (x \cup y) \setminus (x \cap y)$. It is k-edge neighboring, if it differs from G by at most k-edges: $|V \oplus V'| + |E \oplus E'| \leq k$.

Definition 5.2.2 (Node neighboring [60])**.** Given a graph $G = (V, E)$, a graph $G' = (V', E')$ is a *node neighboring graph* of G, if it differs from G by exactly one node, i.e., $|V \oplus V'| = 1$ and $E \oplus E' = \{(u, v) \mid u = V \oplus V' \text{ or } v = V \oplus V'\}$.

With this definition of neighborhood, k-edge DP and node DP are defined analogously to DP for relational databases.

The survey [60] also addresses some of the shortcomings of DP, and reference alternative notions and mechanisms. Most notably, the assumption that all records are generally independent, which implies the assumption that exactly one record encapsulates the participation of a given individual. They point out that this is a serious limitation, because real world data in general, but especially graph-data, often exhibit strong correlations between records.

They also survey notions which address (some of) the limitations:

- Dependent Differential Privacy, which introduces the concept of dependence coefficient that quantifies the level of correlation between two records.

- Pufferfish Privacy, which is a framework, in which each probability distribution (over which the privacy notion is defined) corresponds to an attacker's probabilistic beliefs and background knowledge.

- Inferential Privacy models correlated data as a Markov Chain and adds noise proportionally to the correlation.

- Adversarial privacy, which is weaker than DP but gives higher utility.

- Zero-Knowledge Privacy for average degree query and edit-distance queries.

So overall, while SDC on graph data is more fragmented and heterogeneous than its relational data equivalent, we see an analogous trade-off between syntactic notions, that make assumptions about the structure of the adversaries background, and thus provide weaker privacy guarantees, and semantic notions, that make no such assumptions, but likely come with high utility penalties. This motivates us to, in the next step, analyze whether Association Privacy gives better privacy guarantees than existing syntactic graph privacy notions.

5.3. Analysis of Association Privacy against Graph Adversary

Association Privacy has the goal of protecting the relationships between people in a relation. In this section we analyze if this protection goal holds up against graph-adversaries. For this analysis we want to represent these relationships between people as edges in a graph.

For this representation we propose the following canonical transformation from a relation to a graph:

Definition 5.3.1 (Relation to Graph Transformation)**.** Let $R \in X \times Y$ be a relation over n individuals X, and m attributes Y, given in matrix format. We define an intermediary graph $G^i = (V^i, E^i)$ via the following $(n + m) \times (n + m)$ adjacency matrix:

$$A^i := \begin{pmatrix} 0_{m,m} & R^\top \\ R & 0_{n,n} \end{pmatrix}$$

where $0_{i,i}$ is the matrix of dimension $i \times i$ with 0 for all entries. This graph contains vertices representing individuals, and vertices representing attributes It has an edge between a vertex representing an individual and a vertex representing an attribute, if that individual has that attribute in relation R. Let V_x be the set of vertices representing individuals, and V_y be the set of vertices representing attributes. We construct the final graph $G = (V_x, E)$ by creating an edge between two vertices, if the individuals shared an attribute:

$$(i, j) \in E \text{ if } \exists k \in V_y \text{ with } (i, k) \in E^i \wedge (j, k) \in E^i \,.$$

The intuition behind this transformation is, that since association is assumed when individuals share an attribute, we create an intermediary graph, where individuals are connected to their attributes, and then a final graph, in which individuals share an edge, if they were connected to the same attribute.

Example 5.3.2 (Transforming Relation to Graph)**.** Consider the anonymized Relation S' as given in [33, p. 10], shown in Table 5.1, where • shows the entry added to prevent an adversary from making a specific inference.
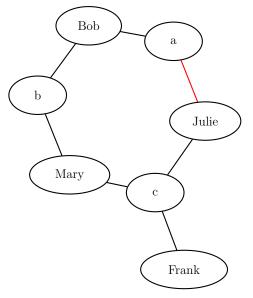
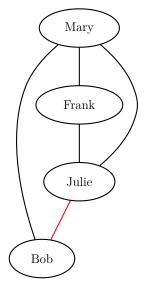| S'	a	b	c
BOB	•	•	
MARY		•	•
FRANK			•
JULIE	•		•

Table 5.1.: Relation S' with Association Privacy

We convert this relation into the intermediary and final graph representations (Figure 5.2), by first converting it into an adjacency matrix (Table 5.2), creating the intermediary graph, shown in Figure 5.2a, and subsequently "dissolving" vertices representing attributes, resulting in the final graph shown in Figure 5.2b.

$A_{S'}$	a	b	c	BOB	MARY	FRANK	JULIE
a				•			•
b				•	•		
c					•	•	•
BOB	•	•					
MARY		•	•				
FRANK			•				
JULIE	•		•				

Table 5.2.: Adjacency matrix corresponding to S'

(a) Intermediary graph representing relation S' with individuals and attributes as vertices, and edges representing that pair being in the relation

(b) Final graph representing S' with individuals as vertices, and edges as associations between those individuals

Figure 5.2.: Graphs Representing Relation S'

5.3.1. Link Disclosure

A definition for link privacy given by [96], is cited by surveys [93, 1]. This definition assumes, that in link privacy the goal is to hide existing links, not the absence of them.

Definition 5.3.3 (Link Privacy (Breach) [96])**.** Given a graph G with adjacency matrix $A = (a_{ij})_{n \times n}$ and the corresponding released graph $\tilde{G}$ with adjacency matrix $\tilde{A} = (\tilde{a}_{ij})_{n \times n}$, use $P(a_{ij} = 1)$ to denote adversaries prior belief about the event of $a_{ij} = 1$ and $P(a_{ij} = 1 | \tilde{G})$ to denote their posterior belief about $a_{ij} = 1$. The released graph $\tilde{G}$ *breaches link privacy,* if

$$P(a_{ij} = 1 | \tilde{G}) > P(a_{ij} = 1).$$

It is of note that unlike the attribute-linkage attack-model for relational data, this definition does not assume that an attacker has only information about the target individual, but also about other individuals and their connections to each other. This is highlighted by k-Anonymity notions, where the attacker has information about connected nodes, or even an entire subgraph.

One issue with this definition is that assumes that some potential targets identity is already known, since knowing link a_{ij} is not valuable information to an attacker, if they do not know who i or j are.

It is not clear if [96] assume pseudo-anonymized or "idendity-labeled" graphs, however their datasets they use lead us to assume, that they work with node-labeled data.

This is not to say a breach in link privacy requires identity disclosure. There are examples in which there are multiple candidate nodes for a given target and all of these candidates are connected to the same neighbor-nodes. So link privacy can be breached without identity disclosure.

For completeness both cases could be analyzed separately: Link privacy given identity-labeled but otherwise anonymized graphs and link privacy given unlabelled anonymized

graphs. However, since identity-labeled graphs only increase an adversary's advantage, we will not consider them, when we already have a negative result without the identity-labels.

We will start our analysis from the strongest adversary: The adversary which has the entire original graph except for one edge as background knowledge. We define two levels of link disclosure, that can occur when link privacy is breached. These act as refinements of Definition 5.3.3.

Definition 5.3.4 (Probabilistic Link Disclosure)**.** Given a graph $\mathcal{G} = (V, E)$ and an anonymized version of that graph $\mathcal{G}' = (V', E')$. Let $\mathcal{G}^* = (V, E^*)$ be the background knowledge of an adversary with $E^* = E \setminus \hat{e}$ for an $\hat{e} \in E$. Assume the adversary knows the identities in V, however V' does not include identity labels.

Probabilistic link disclosure occurs, iff the attackers confidence for the existence of $\hat{e}$ is grater than $1/2$:

$$P(\hat{e}|G') \in (1/2, 1] .$$

This definition is compatible with

Definition 5.3.5 (Link Disclosure without plausible deniability)**.** With the same parameters as in the above Definition if probabilistic link disclosure occurs, we say the target does not have plausible deniability, iff

$$\exists e \colon \Pr[e \in G] = 1 \wedge \Pr[\text{target is node} \neq i] = 0 .$$

We want to address one of Association Privacy's assumptions and how this factors into the analysis: *Observational Monotonicity.* Recall that this assumption meant that if an attribute cannot be observed for an individual, the individual might still have that attribute. If we interpret this in the graph context as "for any given node any non-existent edge to any other node could potentially exist; it just has not been observed", we end up with a graph in which every possible edge exists, and loose all ground for a privacy analysis. This also would be a very powerful, and not very realistic assumption, since clearly a published graph reveals some amount of information. Instead we point out that observational monotonicity applies to information available to the analyst. Together with the completeness assumption we have an analyst, who captures all observable information. From the completeness assumption follows, that an adversary does not have more information as background knowledge, than the analyst who publishes the data. Which is a fact we can exploit in our analysis.

An example that [33, p. 8] gives for observational monotonicity is an individual for which we do not observe that they have cancer but for whom we might yet observe to have cancer. An adversary that knows their target has cancer will not take this individual into consideration, as the adversary knows that their target not only has cancer but also that it can be observed, and thus the analyst must have observed it too. Likewise an adversary that knows cancer cannot be observed for their target, can take this individual in consideration and will discard individuals for which cancer can be observed.

In the following example we will take a look at what happens when the data published is equal to the ground truth, because the relation already fulfilled the notion. While this an unlikely scenario in the real world, it still gives insight of the protections given by the notion.

Examples 5.3.6 (Association Pricacy does not Prevent Link disclosure)**.** Consider relation $R_{5.3.6}$ (Table 5.3). The fact that $R_{5.3.6}$ has Association Privacy can be proven by its

association complex, pictured in Figure 5.3. Observe that all maximal faces are edges/lines, and each sub-face (node) is shared between at least two such edges. Thus the association complex of $R_{5.3.6}$ has no free faces, and the relation has Association Privacy.
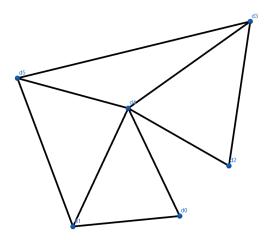

Figure 5.3.: Association Complex of $R_{5.3.6}$

Translating the relation in a graph G as previously described results in the graph shown in Figure 5.4a. Say an adversary has subgraph background knowledge G_{BG}, as shown in Figure 5.4b. The edge unknown to the attacker has been marked red in Figure 5.4a.

$R_{5.3.6}$	a54	a53	a51	a40	a43	a42	a41	a32	a10
0				•					•
1			•				•		•
2						•		•	
3		•			•			•	
4	•			•	•	•	•		
5	•	•	•						

Table 5.3.: Relation $R_{5.3.6}$ with Association Privacy

With this background knowledge link-disclosure without plausible deniability occurs: The attacker can make the following inferences:

- Frank has to be d4, since Frank degree ≥ 4 (the background knowledge is accurate but incomplete) and d4 is the only node with a degree ≥ 4.

- Next Bob, John and Julie have at least degree 3: they occupy some permutation of d1, d3 and d5, since those are the remaining nodes with degree ≥ 3.

- Since the last two remaining nodes, d2 and d0 have the same degree and there are no further restricting indicators Alice and Mary have to be some permutation of d2 and d0.

Even though it is not clear if Alice is node d0 or node d2, the Adversary can still make the inference that Alice shares an edge with Frank and thus link disclosure has occurred.

Even with reduced background knowledge, the adversary is still able to make this inference. Consider the background knowledge G'_{BG} shown in Figure 5.5a. The adversary

(a) $R_{5.3.6}$ As a Graph
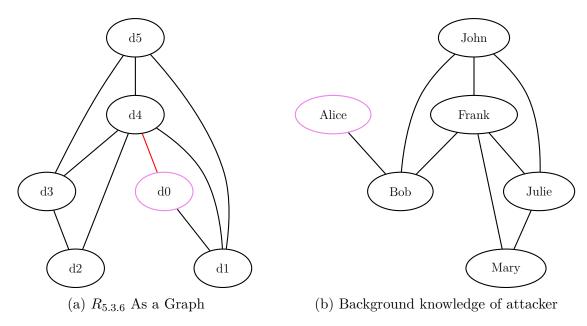
(b) Background knowledge of attacker

Figure 5.4.: Graphs for Examples 5.3.6

can infer that Frank has to be d4 since it is the only node with a sufficient degree. Next, since Bob shares a link with Frank and Alice shares a link with Bob, Alice and Bob have to be one of $(d0, d1), (d1, d5), (d5, d3)$ or $(d3, d2)$. In all of those cases, the adversary can infer that Alice and Frank share an edge.

If we reduce the background information further to G''_{BG} (shown in Figure 5.5b) probabilistic link-disclosure occurs: There are more possibilities for Franks node as its degree does not constrain it anymore to just node d4. The following systematization demonstrates the link disclosure.

- Frank is node d4: Edge Alice-Frank is leaked, see above
  - Bob (connected to Frank) is d0 →Alice (connected to Bob) is d1: Alice shares an edge with Frank ✓
  - Bob is d1
    * Alice is d0: Edge Alice-Frank ✓
    * Alice is d5: Edge Alice-Frank ✓
  - Bob is d5 →Alice is d1 or d3: Edge Alice-Frank ✓✓
  - Bob is d3 →Alice is d5 or d2: Edge Alice-Frank ✓✓
  - Bob is d2 →Alice is d2: Edge Alice-Frank ✓

- Frank is d1
  - Bob is d0 →Alice is d4 ✓
  - Bob is d4
    * Alice is d0 ✓
    * Alice is d2 ✗
    * Alice is d3 ✗
    * Alice is d5 ✓

- Bob is d5
  * Alice is d4 and shares an edge with Frank ✓
  * Alice is d3 and does not share and edge with Frank ✗

- Frank is d3
  - Bob is d2 Alice is d4 ✓
  - Bob is d4 →Alice is...d0 ✗, d1✗, d2✓, d5✓
  - Bob is d5 →Alice is...d1✗, d4 ✓

- Frank is d5
  - Bob is d1 →Alice is...d0✗,d4✓
  - Bob is d3 →Alice is...d2 ✗, d4✓
  - Bob is d4 →Alice is...d0 ✗, d1✓, d2✗, d3✓

So in 20 out of 30 possible worlds Alice and Frank share an edge. Thus link disclosure has occurred even though the Adversary only had background knowledge about four of the nine existing links.

The example can go even further: If the the links John-Frank, and Julie-Frank are removed from the background knowledge, all nodes in the background knowledge have degree 2 or less and thus could be any node in the published graph when going from degree alone. Bot a foothold is be that Alice-Bob-Frank is a path of length 3, and if this path is a circuit, the edge Alice-Frank exists.

Counting all possible such paths reveals there are 17 paths that are not a circuit and 23 paths that are circuits. So the adversary can conclude that the edge Alice-Frank exists even though they only knew about two links.

Only when removing the second to last link is is not possible anymore to decode any other link information. This is because for every possible Alice-Frank link the same link is equally as likely to be a Bob-Frank link instead and vice versa.

The prior examples were concerned with the case in which the data considered is already fulfilling the notion. For a more realistic setting however we have to consider the case in which the published dataset is result of some anonymization mechanism for a dataset which did not initially fulfil the notion. We will however assume that the minimal amount of changes have been made.

Proposition 5.3.7. *Given a Relation R with Association Privacy, the corresponding graph is vulnerable to link disclosure.*

Proof. We prove this by counterexample: Let $R_{5.3.7}$ as defined in Table 5.4 be the output of a mechanism, that ensures Association Privacy (by adding fake properties) with the fewest changes to the to the original relation. Figure 5.6 shows the corresponding graph $G_{5.3.7}$ (Figure 5.6a) and the adversary's background knowledge G_{BG} (Figure 5.6b). The adversary knows that for every edge, there is some likelihood, that it is "fake", meaning it was introduced by the mechanism, as opposed to it being a part of the original dataset. However the adversary can use the knowledge that the amount of changes is minimal. Additionally they know that their background knowledge resembles a subset of the publisher's observations. Figure 5.7 shows the association complex of this relation. We observe that it has Association Privacy, since it has no free faces.

(a) G'_{BG} contains five of nine edges

(b) G''_{BG} contains four of nine edges

(c) G'''_{BG} contains two of nine edges

Figure 5.5.: Reduced background knowledge of adversary

$R_{5.3.7}$	a54	a53	a51	a40	a43	a42	a41	a32	a10	a50
0				•					•	•
1			•					•	•	
2						•		•		
3		•			•			•		
4	•			•	•	•	•			
5	•	•	•							•

Table 5.4.: Output relation $R_{5.3.7}$ with Association Privacy

(a) Graph $G_{5.3.7}$ corresponding to relation $R_{5.3.7}$

(b) Corresponding background knowledge graph G_{BG} of adversary

Figure 5.6.: Graphs associated with relation $R_{5.3.7}$

Figure 5.7.: Association Complex of $R_{5.3.7}$

Starting with the nodes with the highest degree we want to check if they could feasibly be fake. Staring with a50: When removi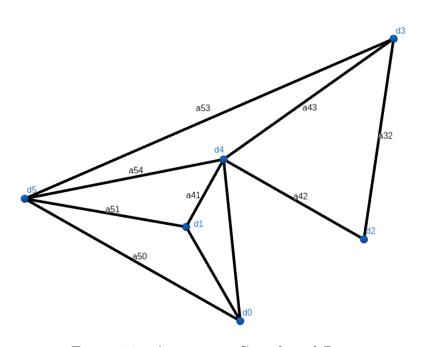ng either or both occurrences of this property, the relation retains Association Privacy. Observe that in the association complex(Figure 5.7) the two affected faces d0 and d5 still are child-faces of two or more maximal faces. This means that the occurrences of this property have to be real, otherwise the assumption of minimal changes would be violated. The same can be said about the occurrences of properties a51, a52, a53, a54, a40, a41, a43, a10 It is however not the case for a42. This can also be observed via the association complex: Removing the face a42 leaves face d2 to be child of only one maximal face (a32). This is the only attribute/ face for which this is the case.

This means that the edges connected to d5 must be true, and all edges connected to d4 — except for (d4,d2) — must be true as well. Edge (d4,d2) is potentially fake. This results in the conclusion that in the real data d5 and d4 each must have degree at least 4, and thus John and Frank must occupy some permutation of d4 and d5. We analyze both possible assignments separately:

- Case 1: John is d5 and Frank is d4. Alice shares a true edge with John (background knowledge), so Alice has to be either d0, d1, or d3. In each of those cases Alice shares a true edge (deduced by minimal changes assumption) with Frank (d4)

- Case 2: John is d4 and Frank is d5. Again, knowing that Alice and John share an edge can be used to find potential nodes for Alice.
    - Nodes d0, d1, and d3 are have a high likelihood to be Alice's node as we deduced all the respective edges to d4 (John) are not fake. In all of these sub-cases Alice shares an edge with Frank (d5).
    - If edge (d4,d2) is a real edge, Alice could also be node d2. In this sub-case Alice would not share an edge with Frank. However we labeled this edge as potentially fake, so this sub-case is less likely than the previously discussed cases.

This probabilistic edge disclosure occurs and the only plausible deniability remaining for Alice and Frank relies on an edge suspected to be fake. □

5.3.1.1. Comparison with other Existing notions

We now proceed to contrast these findings about Association Privacy not protecting against link disclosure with an existing graph privacy notion, carrying a similar protection goal.

We argue, that Association Privacy offers worse privacy protection, than the existing graph privacy notion *k*-security as defined by [18] (which we summarize in subsubsection 5.1.3.9).

One of the protection goals of *k*-security is protecting edges. Their "LinkInfo Security" condition stipulated that given two subgraphs G_A, G_B of the original data as background knowledge and the published Graph G_k the adversary cannot determine from G_k, G_A and G_B that A and B are linked by a path of a certain length with a probability of more than $\frac{1}{k}$ [18].

We believe, that *k*-security prevents link disclosure as we defined it. However proving this is outside of the scope of this thesis. Instead, to show that that Association Privacy

offers weaker protection than *k*-security, we will show in an example, that it is also vulnerable against the adversary proposed by [18].

Example 5.3.8 (Association Privacy is also vulnerable to k-Security adversary). Assume the same published relation and graph as in Examples 5.3.6, name the graph (Figure 5.4a) G_k. Recall that this relation fulfills Association Privacy. Let G_A and G_B be as shown in Figure 5.8. Like in Examples 5.3.6 the adversary can infer that A has to be node $d4$ since that is the only node with sufficient degree. And since every other node is connected to $d4$, the adversary can also infer, that B has to be connected to A with probability 1.
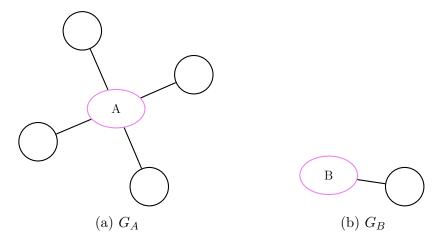

(a) G_A (b) G_B

Figure 5.8.: Background knowledge of k-security adversary

5.3.2. Identity Disclosure

In the previous section we have shown, that Association Privacy does not protect against link disclosure. Which means it fails to deliver on its protection goal of protecting the associations between people. As we noted earlier, there are three main protection goals in graph data privacy: link disclosure, identity disclosure and attribute disclosure. For completeness we present here our intuition, as to why Association Privacy also does not protect against identity disclosure.

We base this intuition on a graph introduced by [1, Fig. 6] which was shown to be vulnerable against identity disclosure. The attack relies on the uniqueness of an attack-subgraph, consisting of four nodes $h_1, \ldots, h_4$ and their connections to the rest of the graph (marked via dashed edges). We present a modified version of this graph G_{idvul} in Figure 5.10. The only modification to the original graph is an additional edge between the nodes 1 and 11 [1].

This added edge does not interfere with the attack, as the attack-subgraph is still unique in the relation. The added edge does not replicate the subgraph, nor does it connect directly to nodes connected with the attack-subgraph. We created a relation R_{idvul} which we show via its simplicial complex in Figure 5.9. This relation has Association

---

[1]It is possible to construct a relation with assocition privacy which corresponds to the original graph. However the relation we were able to find could only be shown to be association private via proof by exhaustion. We chose to instead modify the graph, because for this modified graph we were able to find a corresponding relation that can be shown to be association private directly via its association complex.

Privacy, which can be verified by its association complexes lack of free faces. The relations corresponding graph is the vulnerable graph G_{idvul}.

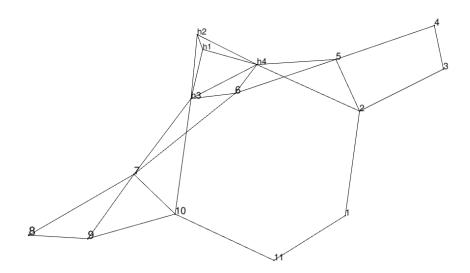This leads to the intuition, that Association Privacy is also vulnerable against identity disclosure.

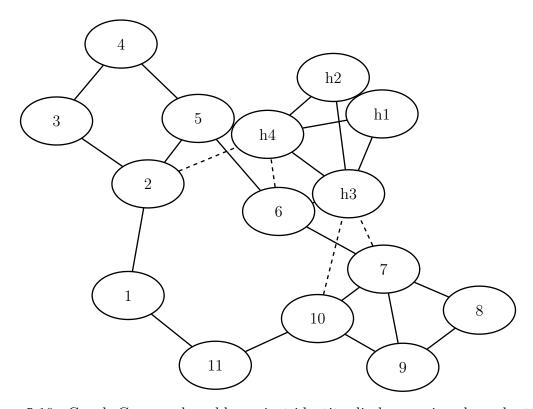Figure 5.9.: Association complex of a relation R_{idvul}

Figure 5.10.: Graph G_{idvul}, vulnerable against identity disclosure via subgraph-attack
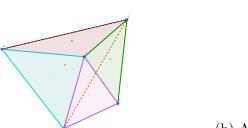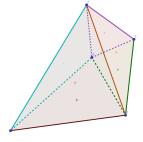
5.3.3. Degree Disclosure

The surveys discussed in section 5.1 agree that the risk beside identity- and link disclosure, is disclosure of some sensitive labels associated with nodes or edges. This is called "content disclosure" by [1, 62], and "attribute disclosure" by [53].

Here we care here about attributes of the node in an unlabelled graphs, such as degree. Not about the attributes associated with a node via labels.

Example 5.3.9. Use relation R of [33, p. 22] as a base and transposes the relation-table, swapping attributes and individuals, the resulting relation R' (Table 5.5) has Association Privacy (since R had Attribute Privacy).

The fact that R' as Association Privacy can also be proven by its association complex. Pictured in Figure 5.11 observe that all maximal faces are triangular, and each sub-face (edge) is shared between exactly two such triangles. Thus the association complex of R' has no free faces, and the relation has Association Privacy.

(a) An angle of the Association Complex of R'

(b) Another angle of the Association Complex of R'

Figure 5.11.: Association Complex of R'

Translating the relation in a graph G as previously described results in the graph shown in Figure 5.12a. Say an adversary has subgraph background knowledge G_{BG} , as shown in Figure 5.12b. The following proofs that the attacker can infer the degree of the target, by enumerating all possible isomorphisms between a subgraph of G and G_{BG}.

The adversary knows the matching node for Alice in the graph must have at least degree 4 since they are in possession of a subgraph. So they can deduce that Alice has to be either node 1,2 or 3 (those are the only nodes with degree 4). Since Alice and Nodes 1,2 or 3 respectively have an edge to all other Nodes, without loss of generality the Adversary can assume Alice is Node 1. For the remaining nodes no deduction based on degree can be made since all remaining nodes in the background knowledge have degree ≤ 3 and all remaining nodes in the released graph have degree of 3 or 4. The remaining nodes of the released graph form the two rings of three ((2, 3, 4) and (2,3,5)), sharing the edge (2,3). So the attacker can deduce that Bob is connected with at least two of Julie, Frank and Mary, providing the information that Bob has at least degree 3.

R'	a	b	c	d	e	f
1	•	•		•	•	
2	•		•	•		•
3		•	•		•	•
4	•	•	•			
5				•	•	•

Table 5.5.: Relation R' with Association Privacy

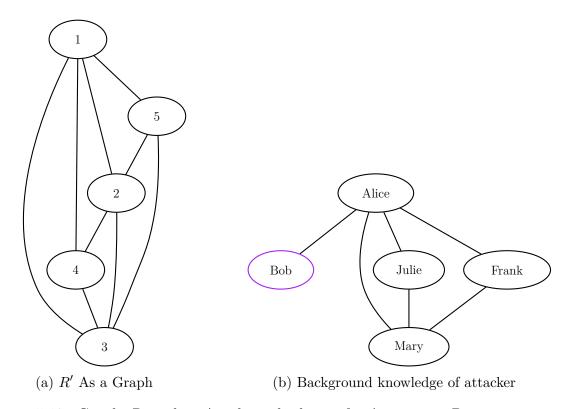(a) R' As a Graph

(b) Background knowledge of attacker

Figure 5.12.: Graphs Revealing Attribute disclosure for Association Privacy

5.4. Association Privacy for Graph Data

In the preceding section we have shown that despite aiming to protect the connections between people, when held up against graph privacy attacks, Association Privacy does not withstand. The next idea might be to analyze which privacy can we achieve, if we take an existing graph, transform it into a relation (that has this graph as a corresponding graph), apply a mechanism achieving one of the topological privacy notions and transform it back into a graph. Unfortunately, there is a problem with this premise: Multiple relations can have the same corresponding graph. Further, relations exist that fulfil either, both or none of Attribute Privacy and Association Privacy respectively, and still all generate the same graph. The triangle graph (Figure 5.13) is a graph for which such relations exist.

Figure 5.13.: Triangle Graph

Example 5.4.1. Given the triangle graph as G, all relations shown in Table 5.6 would result in G, if transformed to a graph. Relation $R_{\mathrm{at,as}}$ fulfills both Attribute Privacy and Association Privacy, $R_{\times,\times}$ fulfills neither. $R_{\mathrm{at},\times}$ fulfills Attribute Privacy, but not Association Privacy, and $R_{\times,\mathrm{as}}$ only fulfills Association Privacy.

$R_{\mathrm{at,as}}$	a	b	c	$R_{\times,\times}$	a	b	c	$R_{\mathrm{at},\times}$	a	$R_{\times,\mathrm{as}}$	a	b	c	d	e	f
1	•		•	1	•			1	•	1	•		•	•		
2	•	•		2	•	•		2	•	2	•	•			•	
3		•	•	3	•	•	•	3	•	3			•	•		•

Table 5.6.: Relations with both, either, or none of Association Privacy and Attribute Privacy

This means, for a given graph G a transformation into a relation (that has G as its corresponding graph) can result in a relation R_G, that already satisfies the desired topological notion, and a mechanism might output R_G unmodified, resulting in the original graph G. However, for the same (now released) graph G a different corresponding relation R'_G can be found, that violates the desired topological notion. This ambiguity in translating a graph into a corresponding relation makes the topological notions fundamentally unsuitable to directly protect data initially given as graph data.

6. Mechanism

In this chapter, we introduce a mechanism we developed to achieve Attribute Privacy. We first explain the intuition behind our approach, then we list the steps involved on a high level. There we also touch on some potential issues, we considered during the design process. Next we introduce the full mechanism, as a set of pseudocode algorithms, and prove in multiple steps, that the overall mechanism always produces relations with Attribute Privacy.

To the best of our knowledge, there does not currently exist any proposed mechanism in the literature that allows to fulfill the topological notions. Correspondingly, no utility analysis exists for these notions. Note that utility in SDC can only be measured for mechanisms achieving the notion, it is not an intrinsic property of any notion.

Example 6.0.1 (Utility is not notion-dependent)**.** The Laplace mechanism [27] is a known DP mechanism. The void mechanism that for all databases outputs the constant number 1 also is DP. While both mechanism satisfy DP, the latter does not provide any utility since all information is lost, while in the former the probability of a query error bigger than α is bounded by $\mathrm{e}^{-\varepsilon\alpha}$ [27].

We have seen in section 4.2, that Attribute Privacy does provide stronger privacy guarantees than k-anonymity in certain aspects, namely the (lack of an) assumption about the structure of the adversaries background-knowledge, and attribute linkage attacks. So to answer the overarching question on whether attribute-privacy can provide a better privacy-utility-tradeoff than the state-of-the-art, we must still achieve the following intermediary goal: Compare the utility of a mechanism achieving Attribute Privacy with the utility of ε-Differential Privacy. We will do this in chapter 7. However, to do this we first need to develop a mechanism, which is what we present in this chapter.

6.1. Mechanism Design and Motivation

In this section we present the high level idea of our mechanism, and motivate the choices we made.

To design a mechanism, we use the property presented in Lemma 2.6.2: A relation whose attribute complex has no free faces has Attribute Privacy. So if we ensure an absence of free faces in the attribute complex, we can guarantee Attribute Privacy. This is the core idea on which we base our mechanism on. We chose this approach because it lines up with the ideas of the topological notions themselves: If the privacy leakage is represented in properties of a topological space, we should ensure that this topological space does not have those properties.

On the highest level, our approach consists of three steps

- Given a relation, construct the corresponding attribute complex.

- Transform this attribute complex into a simplicial complex without free faces.

- Construct a relation from that new simplicial complex.

However this concept cannot be directly realized as we encounter the following challenge: While a relation always has the same attribute complex, two relations can have the same attribute complex. This means that we cannot simply map from the "clean" attribute complex to a clean relation. So instead we modify the approach to the following:

I Given a relation $R \in X \times Y$, construct the attribute complex Φ_R.

II Create $\Phi_{R'}$ by removing all free faces of Φ_R, keeping track of the set of removed faces REMOVED

III Create R' by modifying R in such a way, that no individual creates one of the faces marked for removal: $\forall \gamma \in$ REMOVED: $\nexists x$ with $\gamma \subseteq Y_x$.

Step I is trivially completed, by applying the definition of the attribute complex (Definition 2.4.2). Next we discuss the motivations and realizations of steps II and III.

6.1.1. Remove Free faces

In this subsection we establish, how we realized step II and why we chose to realize it this way.

One possibility to removing free faces in step II, would be to introduce additional simplices, that ensure, that the free faces are no longer free. However, we found two major downsides to this approach: First, inserting new simplices might create new free faces. These new free faces would need to be addressed by further new simplices. Such a strategy might not terminate. To find a strategy that avoids this problem would pose a great challenge. And more importantly, the new simplices might represent a combination of attributes that is unrealistic (for example combining two mutually exclusive attributes) and an attacker could use that to their advantage. For this reason, we chose the approach of removing free faces via *elementary collapse* (Definition 2.3.9) instead.

We found a large body of work about elementary collapse, however the existing research involving elementary collapse was exclusively focused on the goal of simplifying a simplicial complex, in the sense of reducing the number of simplices as far as possible, while keeping topological properties, that are not relevant for our goal, such us the complex homology. [7, 66, 30, 35, 12, 49, 81, 64]. As a concrete example, Paolini [76] present an algorithm that given a simplicial complex tries to find the sequence of elemental collapses that produce the smallest possible simplicial complex. This means if the given simplex is collapsible (Definition 2.3.9) it would find the sequence of elementary collapses, that reduce it to a single vertex, otherwise it finds the simplicial complex without free faces that has the fewest simplices. We would also like to point to *random discrete Morse theory* [7], where an algorithm iteratively performs elementary collapses on free faces of a simplicial complex, and once it arrives at a simplicial complex without free faces it removes a non-free face which is saved as a *critical face*. This continues until the simplicial complex is fully reduced to a single vertex. The number of critical faces in each dimension give an upper bound to some topological properties of the simplicial complex, and thus an algorithms goal is to encounter as few of these critical faces as possible.

We have the opposite goal of performing as few collapses as possible, since every collapse introduces discontinuities (Definition 2.6.4) when translated back to the relation. Think of

a simplicial complex that is reduced to a single vertex. The corresponding clean relation can only have observations of a single attribute (corresponding to that vertex).

In the terminology of random discrete morse theory our goal would be to encounter a critical face as early as possible (at which point we would stop). This means we cannot rely on existing algorithms for removing free cases, since those would guarantee our worst-case.

We would also like to point out that many problem spaces involving simplicial complexes, even seemingly simple ones, turn out to be quite complex problems to solve. For example the problem of determining if a d-dimensional simplicial complex can be collapsed to a k-dimensional simplicial complex (using a series of elemental collapses) is NP-complete for $d \geq k + 2$ (with exception of the case $d = 2, k = 0$) [76].

Due to the complexity of identifying an optimal approach for step II and the limited references available in the literature, we developed an initial, straightforward approach for this step, considering the scope of the project:

1. Given a simplicial complex Φ, identify a free face $f \in \text{free}(\Phi)$ and its encompassing maximal face $m \in \max(\Phi), f \subsetneq m$. If no free face exists, return Φ.

2. Perform an elementary collapse: $\Phi' := \Phi - f - m$.

3. Repeat with Φ', until no free face exists in Φ'.

This approach will give us a first intuition of the viability of topological notions in practice.

6.1.2. Construct a matching Relation

Here we discuss how we achieve step III, in which we construct relation R' from the original relation R, and the set of removed free faces REMOVED such that $\Phi_{R'}$ is the attribute complex.

To do this we perform the following steps:

1. Initialize variables that will be used throughout the algorithm.

   - Save the facets of $\Phi_{R'}$ as CONSTRAINTS $:= \max(\Phi_{R'})$. This variable stores the facets that we need to make sure that they are created by individuals in the final relation. Otherwise the attribute complex of the final relation would miss some facets and thus be different from $\Phi_{R'}$. Once we know that the attributes of some individuals will not be changed anymore, we can remove the facets that these individuals produce from CONSTRAINTS. We call these simplices *constraints*, because they limit the way in which we are allowed to modify an individual's attributes.

   - Save individuals to process $X^* := X$, and initialize individuals to process later LATER $:= \emptyset$. In step 2 we need to iterate over individuals X^*, and check if they produce any of the removed faces. We then need to "process" these individuals, by modifying their attributes in such a way that they no longer produce the undesired faces. The first time step 2 is executed we want to iterate over all individuals in the relation X, however step 2 might be executed multiple times, and later executions will not iterate over all individuals. For this reason we keep track which individuals we process in a given execution of step 2 via variable X^*. For some individuals the processing in step 2 might not be possible the first

time step 2 is executed. These individuals will be marked for later processing, by being inserted in the LATER set.

- Initialize R' as $R' := R$. At the end of the algorithm, R' will contain the attribute private relation.

- Initialize $X^+ := \emptyset$. This set represents fake individuals that were introduced by the algorithm.

2. For every face f in the set of removed free faces: $\forall f \in$ REMOVED

   a) Iterate over all individuals $x \in X^*$ that have all of the attributes from the free face: $\forall x \in \{x \in X^* \mid Y_x \subseteq f\}$

      - We want to make sure, that $Y_x \not\subseteq f$. We can achieve this by picking an attribute $a \in f$ and removing it $R' := R' \setminus \{(x, a)\}$.

      - However, it might happen that we choose a "bad" a, see Example 6.1.1

      - So, we first get all of the constraints that this individual produces

      $$\left(\text{CONSTRAINTS}_x := \big\{c \in \text{CONSTRAINTS} \mid c \subseteq Y_x\big\}\right.$$

      - With this we get the set of attributes that we are not allowed to change for this individual:
      $$\text{FIXED}_x := \bigcup_{\gamma \in \text{CONSTRAINTS}_x} \gamma \, .$$

      - Then we check if $f \setminus \text{FIXED}_x \neq \emptyset$

        - If this is the case we choose $a \in f \setminus \text{FIXED}_x$ and remove it:
        $$R' := R' \setminus \{(x, a)\} \, .$$

        - Otherwise we flag x for later processing and move on without changing the attributes of x: LATER := LATER $\cup \{x\}$

3. If we notice that no individuals could be processed (LATER $= X^*$), we loosen the constraints, by introducing a fake individual which produces one of the constraints $\sigma \in$ CONSTRAINTS

   a) Create a new fake individual x_f and add all vertices of σ as attributes of x_f to the relation:
   $$\left(R' := R' \bigcup_{f \in \text{CONSTRAINTS}} \{(x_f, a) \colon a \in \sigma\} \, .\right.$$

   b) Add x_f to the set of fake individuals X^+

4. We already know some facets which will be contained in the attribute complex of the new relation. Namely these are the facets created by individuals that are already fully processed, and the facets created by fake individuals in X^+, since neither of these will be further altered. We remove these from the constraints.

   - DONE := $X \setminus$ LATER
   - FULFILLED := $\Big\{f \in \max(\Phi_{R'}) \mid \exists x \in (\text{DONE} \cup X^+) \text{ with } Y_x \subseteq f\Big\}$
   - CONSTRAINTS := CONSTRAINTS $\setminus$ FULFILLED

5. If LATER $= \emptyset$ we continue to the next step. Otherwise

   • Set $X^* := \text{LATER}$, LATER $:= \emptyset$ (recall that CONSTRAINTS also has been shrunk) and repeat step 2.

6. If there are no remaining unfulfilled facets (CONSTRAINTS $= \emptyset$) we are done and return R' an attribute private relation over individuals X and attributes Y. Otherwise we need to add individuals that have the corresponding attributes (See Example 6.1.2):

   a) For every facet f in CONSTRAINTS create a new fake individual x_f and add all vertices of f as attributes of x_f to the relation:

   $$R' := R' \bigcup_{f \in \text{CONSTRAINTS}} \{(x_f, a) \colon a \in f\}.$$

   b) Add the set of fake individuals from the previous step to X^+, and set $X' := X \cup X^+$. Return R' an attribute private relation over individuals X' and attributes Y.

When we are constructing a relation from a prior relation, a set of free faces to be removed, and a target attribute complex, we need to pay special attention to the maximal faces of the target relation. We will demonstrate this in the following examples.

Example 6.1.1. Consider relation R, as defined in Table 6.1a. Figure 6.1 shows the attribute complex Φ_R (Figure 6.1a) of R, and the attribute complex $\Phi_{R'}$ with no free faces (Figure 6.1a). Now suppose that by the first part of our algorithm the free face $\{A, B\}$ and its parent maximal face $\{A, B, C\}$ have been collapsed, resulting in the attribute complex $\Phi_{R'}$, which has no remaining free faces. We want to construct R' from R in such a way that $\Phi_{R'}$ is the attribute complex of R', optimizing for fewer changes.

It is clear that individual 1, with attributes $Y_1 = \{A, B, C\}$ needs to be modified, to prevent the creation of simplex $\{A, B\}$. Likewise we need to modify individual 2 with attributes $Y_2 = \{A, B\}$. So possible removals could be

$$R'_1 := R \setminus \{(1, A), (2, A)\}$$
$$R'_2 := R \setminus \{(1, B), (2, B)\}$$
$$R'_3 := R \setminus \{(1, A), (2, B)\}$$
$$R'_4 := R \setminus \{(1, B), (2, A)\}$$

However both R'_1 and R'_3 would fail to produce the required edge $\{A, C\}$ seen in $\Phi_{R'}$. In contrast, both R'_2 and R'_4 are valid candidates and have $\Phi_{R'}$ as their attribute complex.
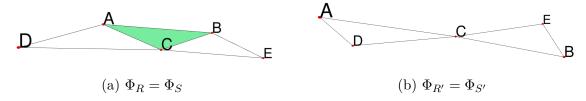

(a) $\Phi_R = \Phi_S$

(b) $\Phi_{R'} = \Phi_{S'}$

Figure 6.1.: Attribute complex Φ_R created from relation R and it's counterpart $\Phi_{R'}$ where free faces have been removed.

R	A	B	C	D	E
1	•	•	•		
2	•	•			
3			•	•	
4	•			•	
5			•	•	
6		•			•
7			•		•

(a) Relation R

S	A	B	C	D	E
1	•	•	•		
2	•	•			
4	•			•	
5			•	•	
6		•			•
7				•	•

(b) Relation S

Table 6.1.: Relations R and S which have the same attribute complex with exactly one edge as a free face

Example 6.1.2. Now consider relation S, as defined in Table 6.1b and its attribute complex Φ_S visualized in Figure 6.1a. Like before we want to reflect the removal of free face $\{A, B\}$ in the relation and have to modify the attributes of individuals 1 and 2 for this.

However, observe that removing attribute A from individual 1 will cause the maximal face $\{A, C\} \in \max(\Phi_{R'})$ to not appear in the corresponding attribute complex. Likewise removing attribute B from individual 1 will cause the maximal face $\{B, C\} \in \max(\Phi_{R'})$ to be missing, since no other individual produces it.

A simple way of rectifying this is to introduce a new individual either producing $\{A, C\}$ [or $\{B, C\}$] and removing either A [or B respectively] from individual 1. In general, when we process individuals, we will mark problematic individuals (such as individual 1 in this example) for later processing. Then we see which of the facets $\max(\Phi_{R'})$ are already produced by individuals we have processed. In some cases this is already enough, but in some cases, like in this one, we end up having to introduce fake individuals which produce the facets required by the cleaned attribute complex.

6.2. Formalizing the Mechanism

Now, that we have established the higher level idea of our mechanism, in this section, we formalize our approach as pseudocode in a series of algorithms. We also provide proofs, that show our mechanism achieves Attribute Privacy.

In Algorithm 1, we define the function `RemoveFreeFaces` (and a helper function `GetFirstFreeFace`), which represents step II of our mechanism. The lines before the first function definition define the steps that are executed if this algorithm is executed by itself. In Algorithm 2, we define the function `RemoveFreeFaceFromRelation`, which realizes step III. And in Algorithm 3 we define a helper function for the prior function, called `AddFakeIndividualToRelation`. Executing Algorithm 2 as a whole represents the entire mechanism, and given a relation R, it creates a relation R' with Attribute Privacy.

Lemma 6.2.1. *Function* `RemoveFreeFaces` *from Algorithm 1 produces a simplicial complex with no free faces.*

Proof. By contradiction: Assume $\exists f \in \text{free}(\Phi')$. This means $\exists m, \in \max(\Phi')$ with $f \subsetneq m$ and $\nexists m' \in \max(\Phi')$ with $f \subsetneq m' \wedge m' \neq m$. So without loss of generality

$$f, m \leftarrow \texttt{GetFirstFreeFace}(\Phi').$$

Algorithm 1: Remove free faces from simplicial complex

Data: Simplicial Complex Φ

Result: Simplicial Complex $\Phi' \subseteq \Phi$, set of removed faces REMOVED

1   $\Phi', \text{REMOVED} := \text{RemoveFreeFaces}(\Phi)$

2   **Function** RemoveFreeFaces(Φ) /* Iteratively removes free faces   */

3   REMOVED $:= \{\}$

4     $\Phi' := \Phi$

5     $f, m := \text{GetFirstFreeFace}(\Phi))$

6   **while** $f \neq \emptyset$ **do**

7       $\Phi' := \Phi' - f - m$

8   REMOVED $:= \text{REMOVED} \cup \{f\}$

9       $f, m := \text{GetFirstFreeFace}(\Phi))$

10   **return** $\Phi', \textit{REMOVED}$

11 **Function** GetFirstFreeFace(Φ')

12   **forall** $m \in \max(\Phi')$ **do**

13   **forall** $f \subsetneq m$ **do**

14   **if** $\nexists m' \in \max(\Phi')$ *with* $f \subsetneq m' \wedge m' \neq m$ **then**

15   **return** f, m

16   **return** $\emptyset, \emptyset$

It follows that the while condition on line 6 will continue to hold true and thus f and m are removed from Φ', which contradicts $f \in \text{free}(\Phi')$. $\qquad\qquad\square$

Proposition 6.2.2. *Given a relation R, and the output of* RemoveFreeFaces$(\Phi_R) =:$ $\Phi', \textit{REMOVED}$ *from Algorithm 1,*

$$R' := \text{RemoveFreeFaceFromRelation}(R, \Phi', \textit{REMOVED})$$

produces a relation R' such that Φ' is the attribute complex of R':

$$\Phi' = \Phi_{R'} .$$

Proof. To prove, that $\Phi_{R'} = \Phi'$ we will first prove, that $\Phi_{R'} \subseteq \Phi'$ and then that $\Phi_{R'} \supseteq \Phi'$.

"$\Phi_{R'} \supseteq \Phi'$": The set CONSTRAINTS is initially filled with the maximal simplices of Φ' (line 5). In line 24 the set FULFILLED is defined with exactly the maximal simplices of Φ', that are produced by individuals in the set DONE. Once an individual is in DONE, their attributes are not further modified by the algorithm. This means these individuals still produce these simplices in the final relation. The same goes for fake individuals ($x \in X^+$), which were added to produce simplices of Φ'. In line 25 these FULFILLED simplices are removed from CONSTRAINTS. And finally in the loop at line 31 individuals producing the remaining attributes are added to the relation. Thus individuals in R' produce all simplices in $\max(\Phi')$, and $\Phi_{R'} \supseteq \Phi'$

"$\Phi_{R'} \subseteq \Phi'$": R' produces no superfluous simplices.

Aside from invocations of AddFakeIndividualToRelation, RemoveFreeFaceFromRelation only removes attributes from R. And AddFakeIndividualToRelation only ever gets invoked with simplices $\sigma \in \text{CONSTRAINTS} \subseteq \Phi'$, and adds an individual producing

---

Algorithm 2: Remove free faces from relation

Data: Relation $R \in X \times Y$

Result: Relation $R' \in X' \times Y$

1 $\Phi_R := \{\gamma \subseteq Y \mid \forall y \in \gamma \exists x \in X : (x, y) \in R\}$ `/* Using definition */`

2 $\Phi', \text{REMOVED} := \texttt{RemoveFreeFaces}(\Phi_R)$

3 $R' := \texttt{RemoveFreeFaceFromRelation}(R, \Phi', \textit{REMOVED})$

4 **Function** $\texttt{RemoveFreeFaceFromRelation}(R, \Phi', \textit{REMOVED})$

5       $\text{CONSTRAINTS} := \max \Phi_R)$

6       $X^* := X$ `/* Individuals to still process           */`

7       $R' := R$ `/* The new relation                     */`

8       $X^+ := \emptyset$ `/* Fake individuals that were added        */`

9   **repeat**

10           $\text{LATER} := \emptyset$

11   **forall** $f \in \textit{REMOVED}$ **do**

12   **forall** $x \in \{x \in X^* \mid Y_x \subseteq f\}$ **do**

13                   $\text{CONSTRAINTS}_x := \{c \in \text{CONSTRAINTS} \mid c \subseteq Y_x\}$

14                   $\text{FIXED}_x := \bigcup\limits_{\gamma \in \text{CONSTRAINTS}_x} \gamma$

15   **if** $f \setminus \textit{FIXED}_x \neq \emptyset$ **then**

                     `/* Vary the choice of a over multiple iterations when`
                          `drawn from the same set                     */`

16                       $a \in f \setminus \text{FIXED}_x$

17                       $R' := R' \setminus \{(x, a)\}$

18   **else**

19                       $\text{LATER} := \text{LATER} \cup \{x\};$

20   **if** $\textit{LATER} = X^*$ **then** `/* No individuals were changed           */`

21               $\sigma \in \text{CONSTRAINTS}$

22               $R', X^+ := \texttt{AddFakeIndividualToRelation}(R', \sigma, X^+)$

23           $\text{DONE} := X \setminus \text{LATER}$

24           $\text{FULFILLED} := \left\{f \in \max(\Phi') \mid \exists x \in (\text{DONE} \cup X^+) \text{ with } Y_x \subseteq f\right\}$

25           $\text{CONSTRAINTS} := \text{CONSTRAINTS} \setminus \text{FULFILLED}$

26           $X^* := \text{LATER}$

27   **until** $\textit{LATER} \neq \emptyset$

28   **if** $\textit{CONSTRAINTS} = \emptyset$ **then**

29   **return** R'

30   **else**

31   **forall** $\sigma \in \textit{CONSTRAINTS}$ **do**

32               $R', X^+ := \texttt{AddFakeIndividualToRelation}(R', \sigma, X^+)$

33           $X' := X \cup X^+$

34   **return** R'

Algorithm 3: Add fake individual to relation

---

1 **Function** `AddFakeIndividualToRelation`(R', σ, X^+)
  /* Adds fake individual, producing simplex σ */
2   $x_f \notin (X \cup X^+)$ /* New fake individual   */
  /* Adding the fake individual to the set of fake individuals   */
  $X^+ := X^+ \cup \{x_f\}$
  /* Giving x_f attributes so it produces simplex σ */
3   $R' := R' \cup \{(x_f, a) : a \in \sigma\}$
4   **return** R', X^+

---

exactly that simplex. This means no new (maximal) simplices, not already in Φ', are created:

$$\forall \sigma \in \Phi_{R'} : \sigma \in \Phi_R \vee \sigma \in \Phi'.$$

Expressed differently:

$$\Phi_{R'} \subseteq (\Phi' \cup \Phi_R).$$

We know that $\Phi' \subseteq \Phi_R$ (Algorithm 1 only removes simplices). So in order to show that $\Phi_{R'} \subseteq \Phi'$ it remains to show that $\Phi_{R'}$ contains no simplices in $\Phi_R - \Phi'$:

$$\Phi_{R'} \cap (\Phi_R - \Phi') = \emptyset.$$

Assume $\exists \sigma \in \Phi_{R'}$ with $\sigma \in (\Phi_R - \Phi')$. This means σ is either a free face itself that was removed $\sigma \in \text{REMOVED}$, or the maximal face that was removed alongside one of its free faces $\exists \sigma_f \subsetneq \sigma, \sigma_f \in \text{REMOVED}$.

- Case 1 $\sigma \in \text{REMOVED}$. Then any individual producing σ will be modified such that they no longer produce σ (line 17).

  - Any individual producing σ will either be modified in this way immediately or added to the LATER set

  - Individuals in the LATER set will be modified in this way in later iterations, when CONSTRAINTS is smaller.

  - Any individual in LATER will be eventually processed, this is ensured by the loop condition **until**LATER $= \emptyset$ at line 27.

  - The check LATER $= X^*$ at line 20 prevents an infinite loop, by adding a fake individual that fulfills (and thus removes) one of the constraints.

  - Eventually either LATER $= \emptyset$ and the loop ends, or CONSTRAINTS $= \emptyset$.

  - The latter case causes one last loop in which $\forall x : \text{FIXED}_x = \emptyset$ (line 14), and thus no individual gets added to LATER.

- Case 2 $\sigma \notin \text{REMOVED}$, but $\exists \sigma_f \subsetneq \sigma, \sigma_f \in \text{REMOVED}$.

  - Any individual producing σ_f will be modified such that they no longer produce σ_f (see case 1).

  - There cannot be any individual producing σ because any such individual would also produce σ_f by nature of set inclusion.

In both cases no individual produces σ, contradicting the assumption.

With this we have shown that $\Phi_{R'} \supseteq \Phi'$ and $\Phi_{R'} \subseteq \Phi'$. Thus we have proven that $\Phi_{R'} = \Phi'$. $\square$

Theorem 6.2.3. *Given a relation $R \in X \times Y$, Algorithm 2 produces a relation $R' \in X' \times Y(X' \supseteq X)$ with Attribute Privacy.*

Proof. In the first step of the algorithm the attribute complex Φ_R of R is input it `RemoveFreeFaces`. This creates a simplicial complex $\Phi' \subseteq \Phi_R$ that has no free faces, and the free faces that were removed in the process:

$$\Phi', \text{REMOVED} \coloneqq \texttt{RemoveFreeFaces}(\Phi_R)$$
$$\text{free}(\Phi') = \emptyset \quad (\textit{Lemma } 6.2.1).$$

Next the algorithm creates R', by invoking `RemoveFreeFaceFromRelation` with the outputs of the last function. This produces a relation R' such that Φ' is the attribute complex of R':

$$R' \coloneqq \texttt{RemoveFreeFaceFromRelation}(R, \Phi', \text{REMOVED})$$
$$\Phi_{R'} = \Phi' \quad (\textit{Proposition } 6.2.2).$$

So R' has an attribute complex without free faces. This means R' has Attribute Privacy (Lemma 2.6.2). $\square$

6.3. Mechanism Limitations

During development of our mechanism, we utilized uniformly randomly generated datasets of the size 10×20. However when we switched to analyze real data we came across two major issues:

First, we noticed, that the time needed to execute the step of finding and removing free faces from Φ_R increased to an intractable amount, once the number of attributes reaches around 14 (the exact number varies, depending on the underlying data). We estimate that

$$\texttt{GetFirstFreeFace} \in \mathcal{O}(\max(|\Phi_i|) \cdot 2^i \cdot \max(|\Phi_i|)),$$

where i is the dimension of the largest simplex:

$$i = \max_{\sigma \in \max(\Phi_R)} (|\sigma|).$$

This is because of the three nested loops, in which we first iterate over all maximal faces $m \in \max(\Phi_R)$, then over all possible sub-faces of m[1] and finally over all maximal faces besides m. This could potentially be improved, by finding a better way of locating free faces.

Next we noticed, that for many real-world inputs, the attribute complex Φ_R collapsed to a single vertex, which has no data utility. In the following we discuss possibilities of avoiding this total collapse of utility.

Just because a simplicial complex is collapsible, does not mean every sequence of elementary collapses will reduce it to a single vertex [63]. Lofano and Newman [63] define,

---

[1]Which equates to the powerset of m (except for the empty set and m itself).

that a sequence of elementary collapses on a complex Σ *gets stuck* at a complex Σ', if the collapsing sequence reduces Σ to Σ', and Σ' has no free faces. Further, they show, that a collapsing sequence can get stuck at a simplicial complex (that has no free faces) despite the fact, that the original simplicial complex was collapsible [63]. For us, getting stuck as early as possible would be ideal behavior. Lofano and Newman prove the following Theorem:

Theorem 6.3.1 (Existence of collapsing sequence [63, Theorem 1.1]). *For $n \geq 8$ and $d \notin \{1, n-3, n-2, n-1\}$, there exists a collapsing sequence of the simplex on n vertices which gets stuck at a d-dimensional complex on n vertices. Moreover, for $n \leq 7$ and d arbitrary or n arbitrary and $d \in \{1, n-3, n-2, n-1\}$ it is not possible to find a collapsing sequence of the simplex on n vertices which gets stuck at dimension d.*

This means if attribute complex Φ_R is collapsible and the the biggest simplex in Φ_R has 7 or fewer vertices, then every sequence of elementary collapses will arrive at the 0-simplex. As a consequence, relations whose attribute complex has at most 6-simplices are at great risk of achieving zero utility. We additionally note, that when we translate a general database with i attribute-groups into a relation with $j \geq i$ sub-attributes, the attribute complex will only have simplices with at most i vertices, since per construction no single individuals will have two sub-attributes of the same attribute-group. For example if the database has four attribute-groups (age, weight, …) and we created a relation where each of those has three sub-attributes (age $< 17, 17 \leq$ age $< 32, 32 \leq$ age) the biggest simplex still can only have 4 vertices.

Thus in order to avoid collapse to a 0-simplex, we need to create relations with attributes from at least 8 attribute-groups. This stands in contrast with the prior observation of this section, that we struggle computationally, when we try to apply the algorithm to relations with more than 14 (sub-)attributes.

Additionally Theorem 6.3.1 gives us a floor for utility loss when the attribute complex is collapsible: Even with $(n+1)$-simplices with $n \geq 8$, no collapsing sequence will get stuck at a dimension $d \geq n-3$. In other words: Every such $(n+1)$-simplex will be reduced to a simplicial complex of at most dimension $n-4$.

Additionally [50] provide an empirical analysis of the probability of a random series of elemental collapses getting stuck on a d-dimensional simplex. The results show an exponential increase in probability of getting stuck. However these probabilities start extremely low and only reach 1.3% for $d = 20$.

6.3.1. Discarded Approaches

We explored possible mitigations, and possible replacement algorithms, to solve the aforementioned issues, however we ended up discarding these approaches, here we lay out these approaches, and the reason, why we did not end up using them.

Given the issue of random elementary collapses not getting stuck easily described above, we considered the following modification to Algorithm 1: In a given round

- Get the list of *all* free faces $f \in \text{free}(\Phi')$ (and their corresponding maximal faces $m \in \max(\Phi')$).

- "Preview" properties of $\Phi_f = \Phi' - f - m$.

- Calculate heuristic that values a larger number of maximal faces and a lower number of free faces:

$$h := |\text{free}(\Phi_f)| - |\text{max}(\Phi_f)| \,.$$

- Choose the best Φ_f according to this heuristic (lower is better) and continue the next round with that simplicial complex

$$\Phi' := \Phi_{f^*}$$

Unfortunately this alteration leads to even worse (compute/ time) performance. For this reason we did not pursue this alteration further.

Erdmann gives a toy example in [33, p. 30], in which he transforms a specific hypothetical relation C without Attribute Privacy, into a relation F with Attribute Privacy. The approach creates new attributes based on the logical `or` of existing attributes. Specifically, the old attributes are observations of individuals picking a combination of two ice-cream flavors like strawberry-vanilla, and the new attributes are the individual flavors, like strawberry or vanilla. Erdmann states that the central idea of the shown approach is removing free faces, and points out, that

- an issue with logical `or`s is that it is easy to construct an attribute that is always true, which would be of no use.

- It is unclear if the new attributes are grounded in what is actually observable. In such a case in which the new attributes would not be observable "one is forced to deal with relation C as given".

We think these drawbacks make this approach unfit to adapt into a general mechanism.

Another observation made by [33] is, that each logical-`or` combination of attributes describes a topological hole[2] in the attribute complex. However, the problematic cases we encountered were problematic because the attribute complex Φ_R is collapsible. A collapsible simplicial complex has no topological holes [4]. This means that in cases in which the attribute complex is collapsible, such as we encountered, this approach would not be able to change the relation at all.

6.4. Concrete Implementation

In addition to pseudocode, we also provide a concrete implementation of our algorithm, which we make available at https://gitlab.kit.edu/urpyg/thesis-topology-of-privacy in the `Scripts` subfolder. We implemented

- the generation of relations from general databases

- creation of the faces of the attribute complex

- exhaustive check if a relation fulfils Attribute Privacy

- `RemoveFreeFaceFromRelation` (Algorithm 2)

- `AddFakeIndividualToRelation` (Algorithm 3)

---

[2]We omit giving a formal definition of a topological hole, but refer readers to [44, Chapter 2] for an introduction into homology groups, the formal representation of topological holes.

in python, using numpy [43] and pandas [83, 68]. This allowed us efficient access to the datasets and to parallelize many of the operations in Algorithm 2. We implemented Algorithm 1 as a polymake [39, 5] ruleset. We chose polymake for this because it provides an efficient implementation of simplicial complex, including retrieval of maximal faces and deletion of a face in a simplicial complex.

7. Utility

In the previous chapter we defined an algorithm that can ensure Attribute Privacy on a given relation. In this chapter we measure the utility provided by that algorithm over two base datasets and two utility metrics. First we introduce the base databases, how we prepare them for the experiments. Then we introduce the metrics, by which we will measure utility After this we introduce our result, in which we applied Attribute Privacy using our mechanism to the test databases, and measured utility using the utility metrics. We contrast these measurements against the corresponding utility measurements on ε-Differential Privacy. Finally we provide additional interpretation of the results.

7.1. Databases and experiments design

In this subsection we present the databases selected for the study and the experiment design. We first preset the two databases we base our experiments on, by elaborating their attributes and the possible values they can have. After this we discuss how we transformed the databases into relations, and present the final resulting relations we will use for our analysis by their attributes. Next we introduce the utility metrics that we will be using. We will also briefly remark on the implementation of ε-Differential Privacy we decided on using.

To test the utility of our mechanism we selected two tabular databases "Cardiovascular" [85] and "adult" [6]. We will refer to these databases as **cardio**, and **adult** respectively. Both are broadly used in the literature and fulfill most of the requisites and assumptions required for Attribute Privacy:

- Relational Completeness: The databases do not contain any missing entries that could be otherwise observed.

- Observational Monotonicity: Since the databases are not relations, we cannot directly apply what it would mean for an attribute to be absent. However we will address this, when creating relations from the databases.

- Observational Accuracy: The databases provide accurate observations.

- A single record always represents one person.

- Direct identifiers have been removed.

The "Cardiovascular" [85] database, obtained from Kaggle, consists of 70000 records of patient data. It includes 12 attributes, categorized into three feature-types and a target: *Objective* features represent Factual information. *Examination* are results of a medical examination. *Subjective* features are information given by the patient.

The attribute describing the presence or absence of cardiovascular disease has been marked by [85] as the target attribute, which an analyst might aim to predict. The attributes and their feature-types, as well as their data-types, and the short names used

when referring to them are shown in Table 7.1. Two attributes listed in this table have an enumerable type. This means they have a set of possible values, represented by numbers. Representations (for both attributes) are 1: normal, 2: above normal, 3: well above normal.

Attribute Name	Short-Name	Feature-Type	Data-Type
Age	`age`	Objective Feature	int (days)
Height	`height`		int (cm)
Weight	`weight`		float (kg)
Gender	`gender`		categorical
Systolic blood pressure	`ap_hi`	Examination Feature	int
Diastolic blood pressure	`ap_lo`		
Cholesterol	`cholesterol`		enumerated: 1,2,3
Glucose	`gluc`		
Smoking	`smoke`	Subjective Feature	binary
Alcohol intake	`alco`		
Physical activity	`active`		
Presence or absence of cardiovascular disease	`cardio`	Target Variable	

Table 7.1.: Attributes of cardiovascular dataset

The second dataset is the "adult" [6] dataset, obtained from the UCI Machine Learning Repository. It contains data extracted from the 1994 Census database and has 14 feature attributes and one target attribute for 32561 individuals. The attributes and their and their data-types are shown in Table 7.2. This database contains multiple attributes with a categorical type, we list the possible values for the categories below. Of these categories `work-class`, `occupation` and `native-country` have some missing values according to the data curator.

workclass Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.

education Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.

marital-status Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.

relationship Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.

race White, Black, Asian-Pac-Islander, Amer-Indian-Eskimo, Other.

native-country United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary,

Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinidad&To-bago, Peru, Hong, Holland-Netherlands.

Attribute Name	Role	Type
age	Feature	int (years)
workclass		categorical
fnlwgt		int
education		categorical
education-num		int
marital-status		categorical
occupation		
relationship		
race		
sex		binary
capital-gain		int
capital-loss		
hours-per-week		
native-country		categorical
income	Target	binary

Table 7.2.: Attributes of adult dataset

7.1.1. Database pre-processing

We need to translate these databases into relations, to enable us enforcement of Attribute Privacy. Binary attributes can directly be translated into attributes in the relation, if they represent an observable attribute whose absence is not significant. We have one instance, in which the original database has a binary attribute, which we cannot simply translate into a single attribute in the relation: The attribute `sex` in the **adult** dataset. This is because in a relation with only one attribute for `sex`, f.e. `sex_male` the absence of that attribute would be interpreted as the presence of a not-represented attribute `sex_female`, which would make that absence significant. So instead we interpret it as a categorical with possible values {Male, Female}.

We interpret attributes that are not binary as attribute-groups. For each categorical attribute-group, we introduce sub-attributes, each representing one of the possible categories. The same goes for enumerable attribute-groups. For attribute-groups with numeric data-types, we introduce buckets for ranges of possible values of these attribute-groups. For these buckets we had to decide on the number of buckets per attribute b, and where to set the boundaries of the buckets.

We decided on three buckets per attribute-group, $b = 3$, since, as mentioned above, a high number of sub-attributes causes performance issues. However we have kept this value variable in our database-to-relation conversion function. For the buckets borders, one approach would be to get the lowest and highest recorded value of the attribute, and then equally divide the resulting interval in b parts. We call this *equal split*. For example `age` in **adult** has values ranging from 17 to 90, so using three buckets with an equal split we would get buckets $(17, 41])$, $(41, 66])$ and $(66, 90]$.

Another approach, which we will call *balanced split*, is to try to find buckets such that each bucket has approximately the same number of individuals. In the prior example of dividing `age` into equal buckets, the buckets $(17, 41]), (41, 66])$ and $(66, 90]$ contained 19926, 11477 and 1158 individuals each respectively. If we instead base the bucket limits on the value distribution of individuals[1] we get the `age` buckets $(16.0, 31.0], (31.0, 44.0]$ and $(44.0, 90.0]$. These contain 11460, 10740 and 10361 individuals respectively. A nice separation like this is not possible though, if a single value occurs too often. For example in `adult` 92% of individuals have `capital-gain` $= 0$. To detect and deal with those cases we do the following:

- Get the normalized spectrum of each possible value.

- If the most frequent value v^* appears in more than $\dfrac{100}{b}$% of individuals, introduce a sub-attribute just for this most-frequent value.

- If v^* is on either end of the value range, create $b - 1$ balanced buckets for the remaining possible values.

- If v^* is somewhere in the middle create $\lfloor \dfrac{b-1}{n} \rfloor$ balanced buckets respectively for values below and above v^*.

Note, that this assumes there exits at most one such ultra-frequent value. This has been true in practice, but one could extend the presented approach to also handle cases with two or more such values.

In the following we will be referring to the following relations.

cardioEqual Relation generated from the **cardio** dataset, using $b = 3$ buckets and equal splits.

cardioBalanced Relation generated from the **cardio** dataset, using $b = 3$ buckets and balanced splits.

adultEqual Relation generated from the **adult** dataset, using $b = 3$ buckets and equal splits.

adultBalanced Relation generated from the **adult** dataset, using $b = 3$ buckets and balanced splits.

Note, that for time we did not test all of these relation to the same extent, and focused on the balanced relations.

With three buckets, the relation representing **cardio** has 27[2] sub-attributes, and the relations for **adult** has 122 sub-attributes. The high sub-attribute count for **adult** is because the database has many categorical attributes with each a lot of possible values. Some of these values appear extremely seldom. Because of the time complexity limitations discussed in section 6.3, we will not be able to apply Attribute Privacy to the entire relations generated this way. So, instead of applying Attribute Privacy to the full relations, we will create sub-relations, by selecting a sets of attributes, and then apply Attribute Privacy to those sub-relations.

---

[1]Choose the i-th bucket such that all individuals in the i-th b-quantile will fall into it for $i \in [1..b]$

[2]The relation based on equal splits has one attribute fewer, because an equal split of the `ap_hi` results in the intervals $(-166, 5240] < (5240, 10630] < (10630, 16020]$ however no individual has an `ap_hi` value in the interval $(5240, 10630]$, which effectively removes this attribute from the relaiton.

Definition 7.1.1 (Selection-based sub-relation)**.** Given a relation $R \in X \times Y$ and a selection of attributes $\mathcal{S} \subseteq Y$,

$$R_{\mathcal{S}} := \{(x, y) \in R \mid y \in \mathcal{S}\}$$

is a *sub-relation* of R based on the *selection* $\mathcal{S}$. We will also annotate this as

$$R_{\mathcal{S}} := \mathrm{select}(R, \mathcal{S}) \,.$$

We will then ensure Attribute Privacy on these sub-relations separately. Because Attribute Privacy does not generally compose sequentially (Recall Proposition 4.1.5 and Proposition 4.1.6), we will treat each of those sub-relation as a separate experiment run. Correspondingly, we will also test ε-Differential Privacy on these sub-relations.

When choosing which sub-attributes to include we need to avoid the following pitfall: Given a set of all sub-attributes $\sigma_{\mathrm{sub}} := \{a_{s1}, \ldots a_{sn}\}$ created from the same attribute-group, if we include all except for one sub attributes ($\sigma_{\mathrm{sub*}=\sigma_{\mathrm{sub}}\setminus\{a_{s*}\}} \in \mathcal{S} \wedge s^* \notin \mathcal{S}$), we break the assumption of observational monotonicity. This is because the absence of all of the included sub-attributes would be significant in the sense that they would imply the presence of the excluded sub-attribute. We illustrate this idea with an informal example: Imagine a source database has a property for height, and we created a relation that has the buckets short, average and tall. If we select a sub-relation that only has attributes short and average, an individual that was neither short nor average could be assumed to be tall. This would mean that the absence of observing short and average was significant, breaking observational monotonicity.

We aimed to include as many attributes as possible, while still being able to perform the algorithm in a reasonable timeframe. We had to weigh

- including attributes from many attribute-groups (because as elaborated in section 6.3, including attributes from 7 or fewer attribute-groups can cause collapse to a single vertex),

- including multiple sub-attributes from the same attribute-group, since a real world analyst would be interested in the comparison,

- and not including too many attributes because this would too long to compute

against each other.

For the relations **cardioEqual**, **cardioBalanced**, and **adultBalanced**[3], we created three sub-relations each.

In Table 7.3 we show the selection sets based on **cardioEqual**. The first row names the corresponding super-attribute from **cardio**. The subsequent rows each represent one selection set, for example the second row represents

$$\mathcal{S}_{ce1} := \{\texttt{age\_(10785, 15103]}, \texttt{age\_(15103, 19408]}, \texttt{age\_(19408, 23713]},$$
$$\texttt{height\_(55, 120]}, \texttt{height\_(120, 185]}, \texttt{height\_(185, 250]},$$
$$\texttt{gluc\_1}, \texttt{gluc\_2}, \texttt{gluc\_3},$$

---

[3]We did not analyze sub relations based on **adultEqual** for considerations of scope, and because we have reasons to believe, that relations based on equal split would provide worse utility, than the corresponding balanced split relations. The idea is, that with an equal split some attributes will have very few instances, which are likely removed and effectively lead the attribute complex to have even fewer vertices.

smoke, alco, active, cardio}

In Table 7.4, we define the selections based on **cardioBalanced**, and in Table 7.5 we define the selections based on **adultBalanced**. We define the sub-relations accordingly, for $i \in \{1, 2, 3\}$:

$$R_{cei} := \text{select}(\textbf{cardioEqual}, \mathcal{S}_{cei})$$
$$R_{cbi} := \text{select}(\textbf{cardioBalanced}, \mathcal{S}_{cbi})$$
$$R_{abi} := \text{select}(\textbf{adultBalanced}, \mathcal{S}_{abi}).$$

These are the nine relations that we will base our main experiments on.

	age			height			weight			cholesterol			gluc			smoke	alco	active	cardio	ap_lo			ap_hi		gender	
S_{ce1}	(10785,15103]	(15103,19408]	(19408,23713]	(55,120]	(120,185]	(185,250]							1	2	3	smoke	alco	active	cardio							
S_{ce2}	(10785,15103]	(15103,19408]	(19408,23713]		(120,185]		(10,73]			1	2	3	1	2	3	smoke	alco	active	cardio						1	2
S_{ce3}	(10785,15103]	(15103,19408]	(19408,23713]	(55,120]	(120,185]	(185,250]	(10,73]	(73,137]	137,200]	1	2	3								(−81,3620]	(3620,7310]	(7310,11000]	(−166,5240]	(10630,16020]	1	2

Table 7.3.: Sub-relation selections based on **cardioEqual**

	age			height			weight			cholesterol			gluc			smoke	alco	active	cardio	ap_lo			ap_hi		gender	
S_{cb1}	(10797,18367]	(18367,20950]	(20950,23713]	(54,160]	(160,168]	(168,250]							1	2	3	smoke	alco	active	cardio							
S_{cb2}	(10797,18367]	(18367,20950]	(20950,23713]		(160,168]		(9,67]			1	2	3	1	2	3	smoke	alco	active	cardio						1	2
S_{cb3}	(10797,18367]	(18367,20950]	(20950,23713]	(54,160]	(160,168]	(168,250]	(9,67]	(67,78]	78,200]	1	2	3								(−70,80]	80	(80,11000]	(−150,120]	120	(120,16020]	1 2

Table 7.4.: Sub-relation selections based on **cardioBalanced**

	age			workclass									fnlwgt			education-num			Race	50K
S_{ab1}	(16,31]	(31,44]	(44,90]	Private	Self-emp-inc	Self-emp-not-inc	Federal-gov	State-gov	Local-gov	Without-pay	Never-worked	?	(12284,141067]	(141067,210474]	(210474,1484705]					
S_{ab2}	(16,31]	(31,44]	(44,90]	Private	Self-emp-inc	Self-emp-not-inc	Federal-gov	State-gov	Local-gov	Without-pay	Never-worked	?	(12284,141067]	(141067,210474]	(210474,1484705]	(0,9]	(9,10]	(10,16]		
S_{ab3}	(16,31]	(31,44]	(44,90]	Private	Self-emp-inc	Self-emp-not-inc	Federal-gov	State-gov	Local-gov	Without-pay	Never-worked	?	(12284,141067]	(141067,210474]	(210474,1484705]				Black White	$>50K$ $\leq 50K$

Table 7.5.: Sub-relation selections on **adultBalanced**

7.1.2. Utility Metrics

We design utility-measurement experiments based on the fixed generalized use cases of histogram publication and micro-aggregation. We have chosen these use cases because they align with common uses cases outlined in [32, 17]. The metrics we use for these use cases are classified as general purpose data metric by [37].

- To evaluate the utility in the histogram publication use case, we use counting queries, as these are commonly used in this case [17]. For these counting queries we will calculate the relative error

$$\text{err} = \frac{Q(D) - Q(\widetilde{D})}{\max\{Q(D), s\}} \, ,$$

  where D and $\widetilde{D}$ respectively are the original and anonymized database, Q a query and s the sanity bound [17, 94], set to 0.1% of the total number individuals in the dataset.

  - A counting query is a query on the database, counting how often a certain attribute appears. For example "how often does `smoke` appear in `cardio`?"
  - We will perform these counting queries for the relations generated from the databases, for both Attribute Privacy and ε-Differential Privacy. This way we can also analyze categorical, numerical and continuous attributes. For example the counting query on the attribute `age_(31, 44]` on **adultBalanced** represents how many people are aged between 31 and 44 years old (right inclusive).
  - We acknowledge, that the boundaries of the buckets in the balanced variants of the relations have not been picked in an differentially private way. However doing so was omitted for reasons of scope, and we believe that the results we received would not have changed noticeably. Instead, we interpret the relations with their inherent bucket boundaries as the input databases to DP.

- For the microdata use case in which the analyst is interested in correlations inherent in the data, we want to measure the distance between the probability distributions of the original and anonymized dataset. To do so we will use distance based on the the Jensen–Shannon Divergence [61, 36], implemented by the `SciPy` [86] pyhon library. It is defined as

$$JS(p, q) \coloneqq \sqrt{\frac{D(p\|m) + D(q\|m)}{2}}$$

  where p and q are the two probability vectors, between which the distance is measured.

  - In our case these are the per-attribute probabilities, that an individual has a given attribute, so the count of that attribute divided by the total number of records. m is the point-wise mean of p and q, and D is the Kullback-Leibler divergence, which we will not define here but refer to [55].
  - We have seen other metrics and divergences used in this context, however we were not able to find a survey comparing the merits and drawbacks of these distances as a utility measure in SDC.
  - We note that [14] states that the commonly used divergences are all special cases of the f-divergence.

7.1.3. Implementation of Differential Privacy

Currently NIST recommends [20] two software libraries for differential privacy:

differential-privacy developed by Google [91], providing libraries in `C++` ,`Go`, and
`Java`, as well as a command-line-interface for differentially private SQL queries. They
provide an implementation of the Laplace mechanism [27], Gaussian mechanism [27],
as well as several thresholding mechanisms [41, 54]. Additionally they provide
utilities for standard statistical operations such as sum, mean, variance, and more,
which they base on the Laplace mechanism.

differential-privacy-library (Diffprivlib) developed by IBM [46], providing a python
library. This library provides implementations of several mechanisms, including
the Laplace mechanism [28], Gaussian mechanism [27], geometric mechanism [40]
and others. The full list of implemented mechanisms can be found in [46, Table
1]. Additionally the library provides differentially private statistical tools mean,
variance and standard deviation, using the Laplace mechanism, as well as histogram
functions using a truncated version of the geometric mechanism.

We chose to use the Laplace mechanism implementation developed by IBM. We chose the
IBM library since it integrates well with our preexisting `numpy` and `pandas` tooling. And
we chose the Laplace mechanism because of its wide usage.

7.2. Results

In this section we present and analyze the results of our utility experiments:

The main goal of this section is to understand the utility implications of our mechanism.
Additionally, to verify our implementation of Algorithm 2 is correct, we implemented Algorithm 4, which verifies that a relation has Attribute Privacy, by exhaustively proving
that for a given relation R, the operator $\phi_R \circ \psi_R$ the identity.

Algorithm 4: Determine has Attribute Privacy

Data: Relation $R \in X \times Y$
Result: Boolean indicating Attribute Privacy

1 **forall** $\gamma \in \mathcal{P}(Y)$ **do** /* powerset   */
2   psi_of_gamma $:= \emptyset$ /* Will contain $\psi_R(\gamma)$ */
3   **forall** $x \in X$ **do**
4   **if** $Y_x \subseteq \gamma$ **then**
5   psi_of_gamma $:=$ psi_of_gamma $\cup \{x\}$
6   phi_after_psi_of_gamma $= Y$ /* Will contain $\phi_R \circ \psi_R(\gamma)$ */
7   **forall** $x \in$ *psi_of_gamma* **do**
8   phi_after_psi_of_gamma $:=$ phi_after_psi_of_gamma $\cap Y_x$
9   **if** *phi_after_psi_of_gamma* $\neq \gamma$ **then**
10   **return** *False*

11 **return** *True*

Plots and charts appearing in this section have been generated using the `matplotlib` [48,
82] python library.

7.2.1. Zero-Utility

In section 6.3, we showed, that attribute complexes with seven or less attribute-groups necessarily collapse to a single vertex, if the attribute complex is collapsible.. This collapse leads to zero-utility. It is not obvious how likely it is for a real world relation with less than seven attribute groups to be collapsible. We conduct an experiment to test, how common it is for a relation with few attributes to collapse to a single vertex. To do this we start with the relation **cardioEqual**, which is based on `cardio` with three buckets and equal splits. We then created all 17875 possible sub-relations with between two and four attributes, and tested, if applying Attribute Privacy would result in a relation where the attribute complex collapsed to a single vertex. We actually noticed, that 11378 (63.65% of all relations) already had Attribute Privacy at the start of the algorithm, so these would not be touched later. Of the remaining 6497 relations, 6067 (93.38%) were reduced to a single vertex.

Likewise on the relation **adultEqual** we performed the same test on 454689 of the sub-relations with between two and four attributes. Out of these 368811 relations (81.11% of all tested) had Attribute Privacy prior to any modifications. Of the 85878 remaining relations 79185 (92.21%) were reduced to a single vertex.

This demonstrates, that in practice, when selecting a low amount of sub-attributes the likelihood of obtaining a relation with a collapsible attribute complex is very high. Because of the low number of attribute-groups that simplicial complex is guaranteed to collapse to a single vertex.

Because of longer compute times for relations with more attributes, we unfortunately cannot perform a similar experiment for relations with more attributes. We would however like to note that with some arbitrary selections of more attributes we found both types of relations: Relations that did collapse to a single vertex, and relations that do not.

Examples 7.2.1. Using `cardio` with three equally split buckets, Consider the sub-relation $R_{\mathcal{S}1}$ with the following 11 attributes

$$\mathcal{S}_1 := \{\texttt{age\_(10785, 15103]}, \texttt{age\_(15103, 19408]}, \texttt{age\_(19408, 23713]},$$
$$\texttt{gender\_2}, \texttt{gender\_1},$$
$$\texttt{height\_(55, 120]}, \texttt{height\_(120, 185]}, \texttt{height\_(185, 250]},$$
$$\texttt{weight\_(10, 73]}, \texttt{weight\_(73, 137]},$$
$$\texttt{gluc\_1}\}.$$

Those attributes come from the 5 attribute-groups `age`, `gender`, `height`, `weight`, and `gluc`. Relation $R_{\mathcal{S}1}$ has a collapsible attribute complex, which necessarily will collapse.

In contrast, consider the sub-relation $R_{\mathcal{S}2}$ from these 11 attributes

$$\mathcal{S}_2 := \{\texttt{age\_(10785, 15103]}, \texttt{age\_(15103, 19408]}, \texttt{age\_(19408, 23713]},$$
$$\texttt{gender\_2}, \texttt{gender\_1},$$
$$\texttt{height\_(55, 120]}, \texttt{height\_(120, 185]}, \texttt{height\_(185, 250]},$$
$$\texttt{weight\_(10, 73]}, \texttt{weight\_(73, 137]}, \texttt{weight\_(137, 200]}\}.$$

The only difference in the two selections is the choice in the last sub-attribute. However despite the small change, the attribute complex of this relation is not collapsible.

7.2.2. Counting Queries

We ensured Attribute Privacy on the sub-relations laid out in subsection 7.1.1 and calculated the mean relative error using all possible counting queries on these relations. We present this result in Table 7.6. There we also note the total number of attributes in each sub-relation, as well as from how many attribute groups they were generated. We note in column *Number of Untouched* the number of attributes for which the corresponding counting query yielded a relative error of 0%. Finally, we note the opposite (how many attributes were completely removed, resulting in a relative error of 100%) in *Number of Removed.*

Input-Relation	Attributes	Attribute-Groups	Mean Relative Error	Number of Untouched	Number of Removed
R_{ce1}	13	7	61.17%	1	6
R_{ce2}	16	9	60.40%	0	7
R_{ce3}	19	7	81.43%	1	15
R_{cb1}	13	7	45.94%	1	2
R_{cb2}	16	9	29.51%	0	0
R_{cb3}	20	7	15.56%	0	0
R_{ab1}	15	3	04.31%	9	0
R_{ab2}	18	4	01.89%	12	0
R_{ab3}	19	5	19.19%	3	2

Table 7.6.: Results of relative error using counting query on relatons that have Attribute Privacy

Next, we calculated the mean relative error for all ε-differentially private counting queries, with

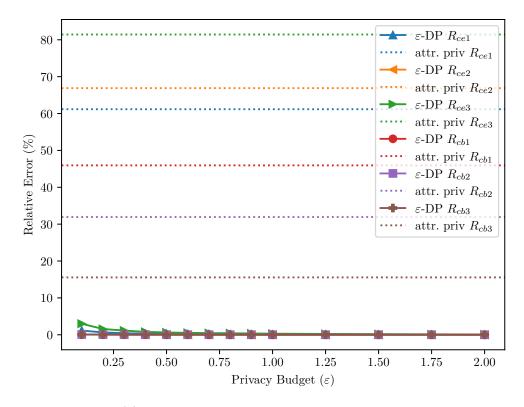$$\varepsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.25, 1.50, 1.75, 2.00\}.$$

We did so, using the Laplace mechanism implemented by IBM's Diffprivlib, on the same sub-relations. For each ε-DP datapoint (specific relation and ε) we calculated the mean relative error for all possible counting queries 100 times, and took the mean of that result.

We present the results, contrasted with the results of Attribute Privacy in Figure 7.1. We observe, that for the relations based on `cardio`, Attribute Privacy delivers much worse utility. For the relations based on `adult`, we observe that Attribute Privacy still delivers worse in most cases, but is closer in relative error. For R_{ab2} Attribute Privacy has a better error rate than the 0.1-differentially private counting queries. However, even for this relation ε-Differential Privacy gives better error rates with $\varepsilon \geq 0.2$.
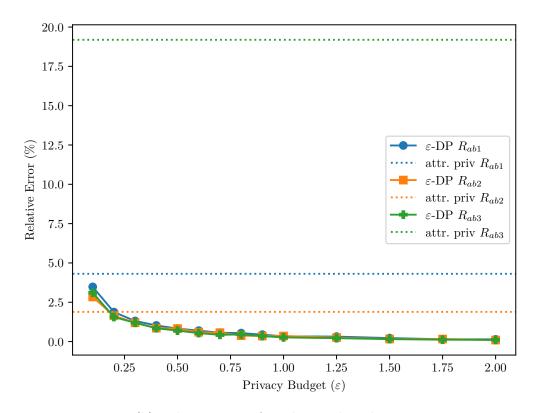
7.2.3. Microdata

For this use case we wanted to measure the distance between the probability vectors with the per-attribute probabilities for the original and private data.

For Attribute Privacy we, like before, ensured Attribute Privacy on the sub-relations laid out in subsection 7.1.1. We denote the attribute variant of a given sub-relation R^*. We then calculated the probability vectors for both the original and the private relations. We denote these per-attribute probability vectors as $v(\cdot)$. We show these distributions in

(a) Relative errors for relations based on `cardio`

(b) Relative errors for relations based on `adult`

Figure 7.1.: Relative error of counting queries on relations under DP, compared with relative error of counting queries on relations protected with Attribute Privacy

comparison with each other in Figure 7.2 and Figure 7.3. The distances we calculated are shown in Figure 7.4 as horizontal lines, together with the corresponding ε-Differential Privacy results.
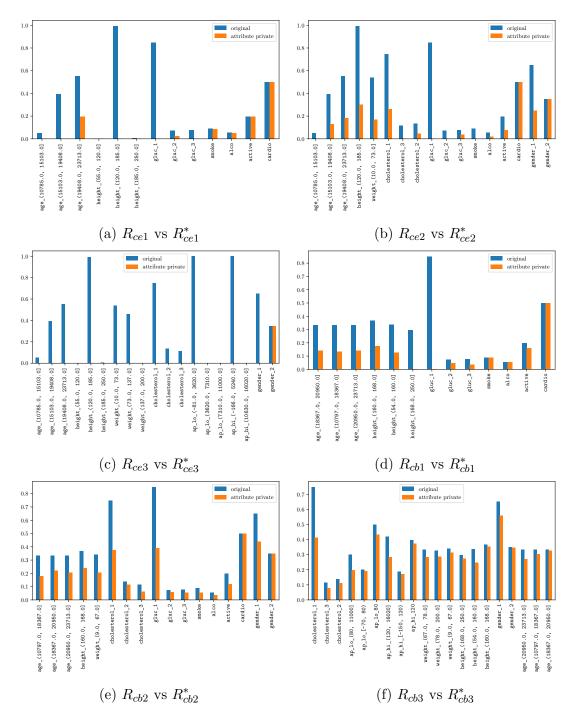

(a) R_{ce1} vs R_{ce1}^*

(b) R_{ce2} vs R_{ce2}^*

(c) R_{ce3} vs R_{ce3}^*

(d) R_{cb1} vs R_{cb1}^*

(e) R_{cb2} vs R_{cb2}^*

(f) R_{cb3} vs R_{cb3}^*

Figure 7.2.: Comparision of probability vecotrs of sub-Relations of **cardioEqual**, **cardioBalanced**, against probability vectors of their corresponding attribute-private relations

For the ε-differentially private probability vector, we need to perform two sequential queries for every attribute in the relation: Let $R \in X \times Y$ be the relation in question and
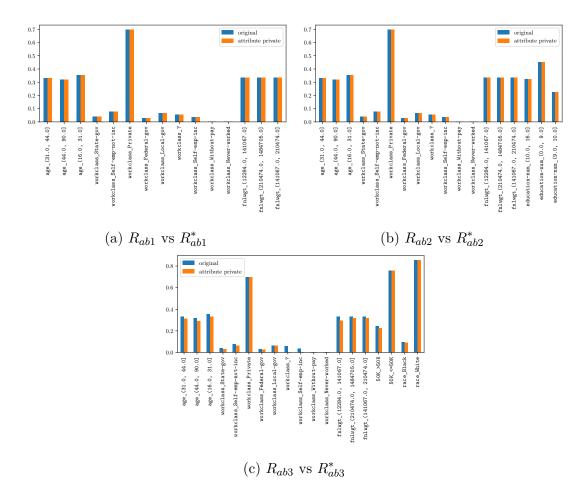
(a) R_{ab1} vs R_{ab1}^*

(b) R_{ab2} vs R_{ab2}^*

(c) R_{ab3} vs R_{ab3}^*

Figure 7.3.: Comparision of probability vecotrs of sub-Relations of **adultBalanced**, against probability vectors of their corresponding attribute-private relations

$n = |Y|$ the number of attributes. For every attribute $a \in Y$ we calculate

$$v_a = \frac{Q\left(a, \frac{\varepsilon}{2n}\right)}{Q\left(n, \frac{\varepsilon}{2n}\right)},$$

where $Q(a, \varepsilon/2n)$ is the $\varepsilon/2n$-differentially private counting query for attribute a, and $Q(n, \varepsilon/2n)$ is the $\varepsilon/2n$-differentially private counting query for the number of attributes. By sequential composition we get the ε-differentially private probability vector

$$v^{dp}(R) = (v_{a_1}, \ldots v_{a_n})^{\top}.$$

We measure the distance to the corresponding (non-private) probability vector $v(R)$. We performed this experiment this with multiple values for ε[4] and the relations laid out in subsection 7.1.1. We repeated this 100 times for each combination of ε and R and averaged the results. In Figure 7.4 we present the final results. Additionally we present the vectors for $\varepsilon = 0.2$ and $R = R_{ab2}$ as a representative example of how both vectors visually compare in Figure 7.5.

We note, that the ε-differentially private results could potentially be improved by using a geometric mechanism [40], instead of performing multiple queries with the laplace mechanism. This (using a geometric mechanism) is also what is done by IBM's Diffprivlib [46, Table 3] when it comes to calculating histogram functions.

Comparing the attribute-private and ε-differentially private results for relations based on `cardio` (Figure 7.4a), we see that the attribute-private results are a lot worse than even the results for 0.1-DP, the smallest tested ε value. In fact, the attribute private results are so much worse, than the corresponding ε-differentially private results, that the differences between different ε values (which range in values between approximately 0.026 for smaller ε values and 0.001 for larger ε values) are contained in the very bottom of the plot. The worst ε-differentially private result in this figure is R_{ce3} for $\varepsilon = 0.1$ with a Jensen-Shannon distance of approximately 0.026. The corresponding (same input relation) attribute private distance is

$$JS(v(R_{ce3}), v(R_{ce3}^*)) \approx 0.7699.$$

The best result for attribute private data in this test suite is

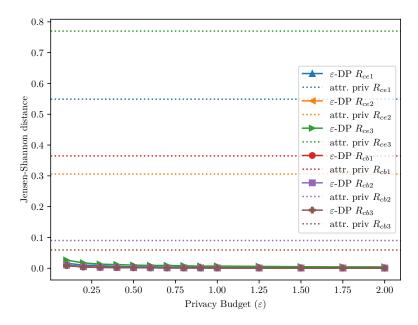$$JS(v(R_{cb3}), v(R_{cb3}^*)) \approx 0.0593,$$

the corresponding DP result is under 0.010 for $\varepsilon = 0.1$ and under 0.005 for $\varepsilon = 0.2$.

The results for the `adult` based relation are notably different. Here the ε-DP curve for R_{ab1} intersects the line for the attribute private result at around $\varepsilon = 1.25$. For R_{ab2}, the attribute private result is better, than all of the tested corresponding ε-DP values. The selection for R_{ab3} differs only in the selection of a few attribute, however the results for this relation are drastically different to the other two:
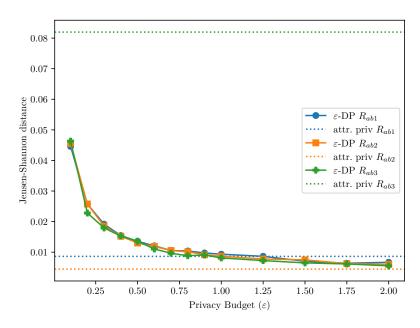
$$JS(v(R_{ab3}), v(R_{ab3}^*)) \approx 0.0820$$

is almost double than the corresponding 0.1-DP distance.

---

[4]$\varepsilon \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.25, 1.50, 1.75, 2.00\}$

(a) JSD for relations based on `cardio`

(b) JSD for relations based on `adult`

Figure 7.4.: Jensen-Shannon Divergence Distance between private and non-private probability vectors for (attribures of) sub-relations $R_{ce1}, R_{ce2}, R_{ce3}, R_{cb1}, R_{cb2}, R_{cb3},$ $R_{ab1}, R_{ab2}, R_{ab3}$, comparing ε-Differential Privacy and attribute private results
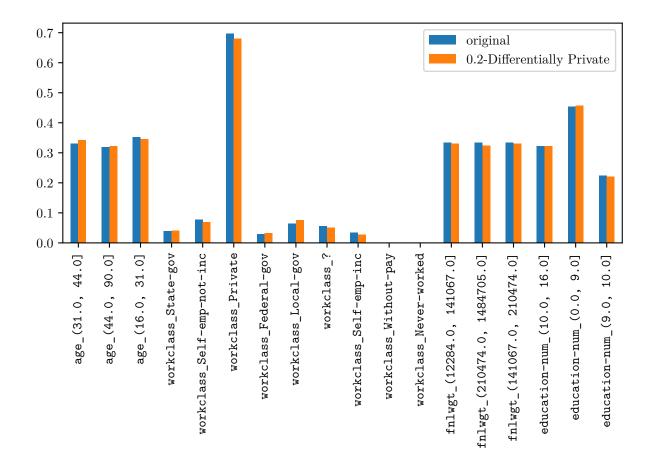
Figure 7.5.: Comparison of 0.2-DP and non-private probability vectors of relation R_{ab2}

7.2.4. Interpretation

In this section we interpret the results gathered in the last two sections.

Under both metrics we have observed, that in most test relations Attribute Privacy provides worse utility than ε-Differential Privacy. For sub-relations based on `cardio` the utility metrics for Attribute Privacy were multiple times worse than their DP counterparts for all tested ε. For the relations based on `adult` Attribute Privacy generally gave utility results that were closer to the ones given by ε-Differential Privacy. However this was not reliably true for all tested sub-relations. We theorize, that this is caused by certain patterns appearing by chance in the data, which will lead to the collapse of many faces in the attribute complex. One such instance is when simplices with 7 or less vertexes appear in the attribute complex which wil necessarily be removed following Theorem 6.3.1. We observe in Table 7.6, that in over half the tested relations multiple attributes were entirely removed.

We conclude, that for a general real world application the utility provided by our mechanism Attribute Privacy is too unreliable, especially since in many cases the utility metrics are multiple times worse than the corresponding 0.1-DP values. Because of Theorem 6.3.1 we believe, that any other mechanism based on the removal of free faces will encounter the same issues we did.

When it comes to execution time Attribute Privacy performs much worse than ε-Differential Privacy. For R_{ab3} the total time needed to achieve an attribute private relation was approximately 422.21577 seconds, of this time the CPU execution time was 407.79311 seconds. In contrast, calculating the differentially private histogram over the same relation with $\varepsilon = 0.5$ took 0.00406 seconds (0.00400 seconds CPU time). The same calculation took approximately the same time for different values for ε. We did not perform precise time measurements for all of the tested relations, but in general the relations we tested took multiple minutes for Attribute Privacy, and a few milliseconds for DP queries.

We attempted to create an attribute private relation of **cardioBalanced**, however after 2436448 seconds (more than 28 days) we still were not done yet with the step that removes free faces from the relation. Calculating the differential private histogram over the same relation with $\varepsilon = 0.5$ took 0.00525 seconds (and approximately the same tile for other values of ε). Since any algorithm implementing Attribute Privacy will have to solve a complex optimization problem we do not believe computation time even close to ε-Differential Privacy could be achieved by an Attribute Privacy algorithm.

8. Conclusion

From our analysis of Association Privacy in the context of graph data privacy, in chapter 5, we draw the conclusion, that despite Association Privacy's goal of protecting the relationships between people, it is vulnerable against a graph privacy attacker, and the existing syntactic notion of k-security [18], provides stronger protections. Therefore, we discourage the further use of this notion as a privacy metric in the field of graph-data privacy.

When we compared the privacy guarantees of the topological notions with the relational-data state-of-the-art in chapter 4, we found that the topological notions have an advantage over k-Anonymity, when it comes to the assumptions, as they do not assume a known quasi-identifier, like k-Anonymity does. Further, we found that Attribute Privacy protects against attribute-linkage attacks, against which plain k-Anonymity is vulnerable. Additionally, we found that Attribute Privacy does compose in parallel. However, Attribute Privacy is vulnerable against the other tested attacks, including record linkage, which is prevented by k-Anonymity. Association Privacy is vulnerable against all tested attacks. k-Anonymity with ℓ-Diversity does protect against attribute-linkage as well, and t-closeness adds protection against probabilistic attacks. These are multiple advantages, k-Anonymity with extensions have over Attribute Privacy, and diminishes the advantage, attribute-privacy has over the syntactic SoTA to the advantage in assumptions and composition.

We conclude, for this question, that in terms of privacy guarantees, Association Privacy does not provide any privacy advantages over any of the SoTA. Attribute Privacy may be chosen over k-Anonymity and extensions, if the need to forego assumptions about quasi-identifier, outweighs the need for protection against record linkage, and probabilistic attacks. We further conclude, that Attribute Privacy is inferior in attack resilience to DP, therefore the only reason to use Attribute Privacy over DP is if it is able to provide superior utility.

However, our utility analysis shows that Attribute Privacy does not provide better utility than DP. In the process of designing a mechanism for Attribute Privacy (chapter 6), we found, that any approach based on collapsing free faces (like ours does) needs to solve a complex optimization problem, which limits the maximal number of attributes because of computational complexity. At the same time, we showed that a low number of attributes guarantees a high utility loss. Additionally, we argued that potential alternative approaches would have similar flaws. In our utility analysis of our mechanism (chapter 7), we found, that the mechanism for Attribute Privacy provides much worse utility, than the Laplace mechanism for ε-DP, even for very small values of ε. From our previous observations, we conclude, that any generalized mechanism for Attribute Privacy would likely provide similarly poor utility.

Regarding the possible combination of Attribute Privacy and DP to mitigate correlation attacks, we found that Attribute Privacy is vulnerable when input data shows certain correlations. From this we draw, that we cannot combine Attribute Privacy with DP, in order to protect DP against such correlation-based attacks.

Finally, we conclude that while the topological privacy notions showed some initial promise, with their lack of assumptions about quasi-identifiers, further analysis revealed multiple flaws inherent to the notions. In our analysis, we failed to find merit in the protections promised by Association Privacy. There might be some niche applications for Attribute Privacy, but it is not suited as a generalized replacement for the existing relational SoTA, as it does not provide a better privacy-utility tradeoff.

Bibliography

[1] Jemal H. Abawajy, Mohd Izuan Hafez Ninggal, and Tutut Herawan. "Privacy Preserving Social Network Data Publication". In: *IEEE Communications Surveys & Tutorials* 18.3 (2016), pp. 1974–1997. ISSN: 1553-877X. DOI: 10.1109/COMST.2016.2533668.

[2] Rasim Alguliyev, Ramiz Aliguliyev, and Farhad Yusifov. "Graph Modelling for Tracking the COVID-19 Pandemic Spread". In: *Infectious Disease Modelling* 6 (Jan. 1, 2021), pp. 112–122. ISSN: 2468-0427. DOI: 10.1016/j.idm.2020.12.002.

[3] Margo J. Anderson. *The American Census: A Social History*. Yale University Press, 2015. URL: https://books.google.com/books?hl=en&lr=&id=NzNOCgAAQBAJ&oi=fnd&pg=PP9&dq=census&ots=bUVPNNoprg&sig=5FQLyiXh8PHSHe7K6TTOQOAzWJc (visited on 10/10/2024).

[4] Lior Aronshtam et al. "Collapsibility and Vanishing of Top Homology in Random Simplicial Complexes". In: *Discrete & Computational Geometry* 49.2 (Mar. 1, 2013), pp. 317–334. ISSN: 1432-0444. DOI: 10.1007/s00454-012-9483-8.

[5] Benjamin Assarf et al. "Computing Convex Hulls and Counting Integer Points with Polymake". In: *Mathematical Programming Computation* 9.1 (Mar. 1, 2017), pp. 1–38. ISSN: 1867-2957. DOI: 10.1007/s12532-016-0104-z.

[6] Ronny Kohavi Barry Becker. *Adult*. UCI Machine Learning Repository, 1996. DOI: 10.24432/C5XW20.

[7] Bruno Benedetti and Frank H. Lutz. "Random Discrete Morse Theory and a New Library of Triangulations". In: *Experimental Mathematics* 23.1 (Jan. 2, 2014), pp. 66–94. ISSN: 1058-6458, 1944-950X. DOI: 10.1080/10586458.2013.865281.

[8] Bruno Benedetti et al. "Random Simple-Homotopy Theory". In: *Journal of Applied and Computational Topology* (Oct. 28, 2023). ISSN: 2367-1734. DOI: 10.1007/s41468-023-00139-4.

[9] James Bennett and Stan Lanning. "The Netflix Prize". In: *Proceedings of KDD Cup and Workshop*. Vol. 2007. New York, 2007, p. 35. URL: https://www.academia.edu/download/90881302/NetflixPrize-description.pdf (visited on 10/10/2024).

[10] Casper Solheim Bojer and Jens Peder Meldgaard. "Kaggle Forecasting Competitions: An Overlooked Learning Opportunity". In: *International Journal of Forecasting* 37.2 (Apr. 1, 2021), pp. 587–603. ISSN: 0169-2070. DOI: 10.1016/j.ijforecast.2020.07.007.

[11] Paolo Boldi et al. *Injecting Uncertainty in Graphs for Identity Obfuscation*. Aug. 20, 2012. DOI: 10.48550/arXiv.1208.4145. arXiv: 1208.4145 [cs]. Pre-published.

[12] Benjamin A. Burton et al. "Parameterized Complexity of Discrete Morse Theory". In: *ACM Trans. Math. Softw.* 42.1 (Mar. 1, 2016), 6:1–6:24. ISSN: 0098-3500. DOI: 10.1145/2738034.

[13] Ji-Won Byun et al. "Secure Anonymization for Incremental Datasets". In: *Secure Data Management*. Ed. by Willem Jonker and Milan Petković. Berlin, Heidelberg: Springer, 2006, pp. 48–63. ISBN: 978-3-540-38987-3. DOI: 10.1007/11844662_4.

[14] Yuhang Cai and Lek-Heng Lim. *Distances between Probability Distributions of Different Dimensions*. Jan. 29, 2022. arXiv: 2011.00629 [cs, math, stat]. URL: http://arxiv.org/abs/2011.00629 (visited on 03/25/2024). Pre-published.

[15] Yang Cao et al. "Quantifying Differential Privacy in Continuous Data Release Under Temporal Correlations". In: *IEEE Transactions on Knowledge and Data Engineering* 31.7 (July 2019), pp. 1281–1295. ISSN: 1558-2191. DOI: 10.1109/TKDE.2018.2824328.

[16] Jordi Casas-Roma et al. "K-Degree Anonymity on Directed Networks". In: *Knowledge and Information Systems* 61.3 (Dec. 1, 2019), pp. 1743–1768. ISSN: 0219-3116. DOI: 10.1007/s10115-018-1251-5.

[17] Rui Chen et al. "Publishing Set-Valued Data via Differential Privacy". In: *Proceedings of the VLDB Endowment* 4.11 (June 3, 2020), pp. 1087–1098. ISSN: 2150-8097. DOI: 10.14778/3402707.3402744.

[18] 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. New York, NY, USA: Association for Computing Machinery, June 6, 2010, pp. 459–470. ISBN: 978-1-4503-0032-2. DOI: 10.1145/1807167.1807218.

[19] Sean Chester et al. "Anonymizing Subsets of Social Networks with Degree Constrained Subgraphs". In: *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. Aug. 2012, pp. 418–422. DOI: 10.1109/ASONAM.2012.74.

[20] David Darais. *Summation and Average Queries: Detecting Trends in Your Data*. NIST. Dec. 17, 2020. URL: https://www.nist.gov/blogs/cybersecurity-insights/summation-and-average-queries-detecting-trends-your-data (visited on 09/01/2024).

[21] Damien Desfontaines and Balázs Pejó. *SoK: Differential Privacies*. Nov. 13, 2022. DOI: 10.48550/arXiv.1906.01337. arXiv: 1906.01337. Pre-published.

[22] J. Domingo-Ferrer and J. Soria-Comas. "From T-Closeness to Differential Privacy and Vice Versa in Data Anonymization". In: *Knowledge-Based Systems* 74 (Jan. 2015), pp. 151–158. ISSN: 09507051. DOI: 10.1016/j.knosys.2014.11.011. arXiv: 1512.05110 [cs].

[23] Josep Domingo-Ferrer and Jordi Soria-Comas. *Connecting Randomized Response, Post-Randomization, Differential Privacy and t-Closeness via Deniability and Permutation*. Mar. 6, 2018. DOI: 10.48550/arXiv.1803.02139. arXiv: 1803.02139 [cs]. Pre-published.

[24] C. Dwork et al. "Pan-Private Streaming Algorithms". In: International Conference on Supercomputing. 2010. URL: https://api.semanticscholar.org/CorpusID:7916594 (visited on 05/30/2024).

[25] Cynthia Dwork. "Differential Privacy". In: *Automata, Languages and Programming*. Ed. by Michele Bugliesi et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2006, pp. 1–12. ISBN: 978-3-540-35908-1. DOI: 10.1007/11787006_1.

[26] Cynthia Dwork. "Differential Privacy: A Survey of Results". In: *Theory and Applications of Models of Computation*. Ed. by Manindra Agrawal et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 1–19. ISBN: 978-3-540-79228-4. DOI: 10.1007/978-3-540-79228-4_1.

[27] Cynthia Dwork and Aaron Roth. "The Algorithmic Foundations of Differential Privacy". In: *Foundations and Trends® in Theoretical Computer Science* 9.3–4 (Aug. 10, 2014), pp. 211–407. ISSN: 1551-305X, 1551-3068. DOI: 10.1561/0400000042.

[28] Cynthia Dwork et al. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography*. Ed. by Shai Halevi and Tal Rabin. Berlin, Heidelberg: Springer, 2006, pp. 265–284. ISBN: 978-3-540-32732-5. DOI: 10.1007/11681878_14.

[29] Cynthia Dwork et al. "Differential Privacy under Continual Observation". In: *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*. STOC '10. New York, NY, USA: Association for Computing Machinery, June 5, 2010, pp. 715–724. ISBN: 978-1-4503-0050-6. DOI: 10.1145/1806689.1806787.

[30] Ömer Eğecioğlu and Teofilo F. Gonzalez. "A Computationally Intractable Problem on Simplicial Complexes". In: *Computational Geometry* 6.2 (May 1, 1996), pp. 85–98. ISSN: 0925-7721. DOI: 10.1016/0925-7721(95)00015-1.

[31] Emelie Ekenstedt et al. "When Differential Privacy Implies Syntactic Privacy". In: *IEEE Transactions on Information Forensics and Security* 17 (2022), pp. 2110–2124. ISSN: 1556-6021. DOI: 10.1109/TIFS.2022.3177953.

[32] Mark Elliot and Josep Domingo-Ferrer. *The Future of Statistical Disclosure Control*. Dec. 21, 2018. DOI: 10.48550/arXiv.1812.09204. arXiv: 1812.09204 [cs]. Pre-published.

[33] Michael Erdmann. *Topology of Privacy: Lattice Structures and Information Bubbles for Inference and Obfuscation*. Dec. 11, 2017. DOI: 10.48550/arXiv.1712.04130. arXiv: 1712.04130 [cs, math]. Pre-published.

[34] *Find Open Datasets and Machine Learning Projects | Kaggle*. URL: https://www.kaggle.com/datasets (visited on 10/10/2024).

[35] Robin Forman. "Morse Theory for Cell Complexes". In: *Advances in Mathematics* 134.1 (Mar. 1, 1998), pp. 90–145. ISSN: 0001-8708. DOI: 10.1006/aima.1997.1650.

[36] B. Fuglede and F. Topsoe. "Jensen-Shannon Divergence and Hilbert Space Embedding". In: *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings*. International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings. June 2004, pp. 31–. DOI: 10.1109/ISIT.2004.1365067.

[37] Benjamin C. M. Fung et al. "Privacy-Preserving Data Publishing: A Survey of Recent Developments". In: *ACM Computing Surveys* 42.4 (June 2010), pp. 1–53. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/1749603.1749605.

[38]   Srivatsava Ranjit Ganta, Shiva Prasad Kasiviswanathan, and Adam Smith. "Composition Attacks and Auxiliary Information in Data Privacy". In: *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* KDD '08. New York, NY, USA: Association for Computing Machinery, Aug. 24, 2008, pp. 265–273. ISBN: 978-1-60558-193-4. DOI: 10.1145/1401890.1401926.

[39]   Ewgenij Gawrilow and Michael Joswig. "Polymake: A Framework for Analyzing Convex Polytopes". In: *Polytopes — Combinatorics and Computation.* Ed. by Gil Kalai and Günter M. Ziegler. Basel: Birkhäuser, 2000, pp. 43–73. ISBN: 978-3-0348-8438-9. DOI: 10.1007/978-3-0348-8438-9_2.

[40]   Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. *Universally Utility-Maximizing Privacy Mechanisms.* Version 3. Mar. 20, 2009. DOI: 10.48550/arXiv.0811.2841. arXiv: 0811.2841 [cs]. Pre-published.

[41]   Michaela Gotz et al. "Publishing Search Logs—A Comparative Study of Privacy Guarantees". In: *IEEE Transactions on Knowledge and Data Engineering* 24.3 (Mar. 2012), pp. 520–532. ISSN: 1558-2191. DOI: 10.1109/TKDE.2011.26.

[42]   Patricia Guerra-Balboa et al. *Composition in Differential Privacy for General Granularity Notions (Long Version).* Apr. 17, 2024. arXiv: 2308.14649 [cs]. URL: http://arxiv.org/abs/2308.14649 (visited on 05/21/2024). Pre-published.

[43]   Charles R. Harris et al. "Array Programming with NumPy". In: *Nature* 585.7825 (Sept. 17, 2020), pp. 357–362. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/s41586-020-2649-2.

[44]   Allen Hatcher. *Algebraic Topology.* 14th printing 2015. Cambridge: Cambridge University Press, 2015. 544 pp. ISBN: 978-0-521-79540-1.

[45]   Michael Hay et al. "Resisting Structural Re-Identification in Anonymized Social Networks". In: *The VLDB Journal* 19.6 (Dec. 1, 2010), pp. 797–823. ISSN: 0949-877X. DOI: 10.1007/s00778-010-0210-x.

[46]   Naoise Holohan et al. *Diffprivlib: The IBM Differential Privacy Library.* July 4, 2019. DOI: 10.48550/arXiv.1907.02444. arXiv: 1907.02444 [cs]. Pre-published.

[47]   Thomas Humphries et al. "Investigating Membership Inference Attacks under Data Dependencies". In: *2023 IEEE 36th Computer Security Foundations Symposium (CSF).* 2023 IEEE 36th Computer Security Foundations Symposium (CSF). July 2023, pp. 473–488. DOI: 10.1109/CSF57540.2023.00013.

[48]   John D. Hunter. "Matplotlib: A 2D Graphics Environment". In: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. ISSN: 1521-9615. DOI: 10.1109/MCSE.2007.55.

[49]   Michael Joswig and Marc E. Pfetsch. "Computing Optimal Morse Matchings". In: *SIAM Journal on Discrete Mathematics* 20.1 (Jan. 2006), pp. 11–25. ISSN: 0895-4801, 1095-7146. DOI: 10.1137/S0895480104445885.

[50]   Michael Joswig et al. "Frontiers of Sphere Recognition in Practice". In: *Journal of Applied and Computational Topology* 6.4 (Dec. 1, 2022), pp. 503–527. ISSN: 2367-1734. DOI: 10.1007/s41468-022-00092-8.

[51] Daniel Kifer and Ashwin Machanavajjhala. "No Free Lunch in Data Privacy". In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data.* SIGMOD/PODS '11: International Conference on Management of Data. Athens Greece: ACM, June 12, 2011, pp. 193–204. ISBN: 978-1-4503-0661-4. DOI: 10.1145/1989323.1989345.

[52] Daniel Kifer et al. *Bayesian and Frequentist Semantics for Common Variations of Differential Privacy: Applications to the 2020 Census.* Sept. 7, 2022. DOI: 10.48550/arXiv.2209.03310. arXiv: 2209.03310 [cs, stat]. Pre-published.

[53] M. Kiranmayi and N. Maheswari. "A Review on Privacy Preservation of Social Networks Using Graphs". In: *Journal of Applied Security Research* 16.2 (Apr. 3, 2021), pp. 190–223. ISSN: 1936-1610. DOI: 10.1080/19361610.2020.1751558.

[54] Aleksandra Korolova et al. "Releasing Search Queries and Clicks Privately". In: *Proceedings of the 18th International Conference on World Wide Web.* WWW '09. New York, NY, USA: Association for Computing Machinery, Apr. 20, 2009, pp. 171–180. ISBN: 978-1-60558-487-4. DOI: 10.1145/1526709.1526733.

[55] S. Kullback and R. A. Leibler. "On Information and Sufficiency". In: *The Annals of Mathematical Statistics* 22.1 (1951), pp. 79–86. ISSN: 0003-4851. JSTOR: 2236703. URL: https://www.jstor.org/stable/2236703 (visited on 09/05/2024).

[56] Kaiyang Li et al. "Plausible Heterogeneous Graph K-Anonymization for Social Networks". In: *Tsinghua Science and Technology* 27.6 (Dec. 2022), pp. 912–924. ISSN: 1007-0214. DOI: 10.26599/TST.2021.9010083.

[57] 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.* 2007 IEEE 23rd International Conference on Data Engineering. Apr. 2007, pp. 106–115. DOI: 10.1109/ICDE.2007.367856.

[58] Ninghui Li et al. "A Primer on -Differential Privacy". In: *Differential Privacy: From Theory to Practice.* Ed. by Ninghui Li et al. Cham: Springer International Publishing, 2017, pp. 7–31. ISBN: 978-3-031-02350-7. DOI: 10.1007/978-3-031-02350-7_2.

[59] Yan Li et al. "Privacy Leakage Analysis in Online Social Networks". In: *Computers & Security* 49 (Mar. 1, 2015), pp. 239–254. ISSN: 0167-4048. DOI: 10.1016/j.cose.2014.10.012.

[60] Yang Li et al. "Private Graph Data Release: A Survey". In: *ACM Computing Surveys* 55.11 (Feb. 22, 2023), 226:1–226:39. ISSN: 0360-0300. DOI: 10.1145/3569085.

[61] J. Lin. "Divergence Measures Based on the Shannon Entropy". In: *IEEE Transactions on Information Theory* 37.1 (Jan. 1991), pp. 145–151. ISSN: 1557-9654. DOI: 10.1109/18.61115.

[62] 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. New York, NY, USA: Association for Computing Machinery, June 9, 2008, pp. 93–106. ISBN: 978-1-60558-102-6. DOI: 10.1145/1376616.1376629.

[63] Davide Lofano and Andrew Newman. *The Worst Way to Collapse a Simplex.* Aug. 13, 2020. arXiv: 1905.07329 [math]. URL: http://arxiv.org/abs/1905.07329 (visited on 12/01/2022). Pre-published.

[64] Wenyu Ma et al. "Simplicial Complex Reduction Algorithm for Simplifying WSN's Topology". In: *Ad Hoc Networks*. Ed. by Jun Zheng et al. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Cham: Springer International Publishing, 2019, pp. 25–35. ISBN: 978-3-030-05888-3. DOI: 10.1007/978-3-030-05888-3_3.

[65] Ashwin Machanavajjhala et al. "L-Diversity: Privacy beyond k-Anonymity". In: *ACM Transactions on Knowledge Discovery from Data* 1.1 (Mar. 1, 2007), 3–es. ISSN: 1556-4681. DOI: 10.1145/1217299.1217302.

[66] Rémy Malgouyres and Angel R. Francés. "Determining Whether a Simplicial 3-Complex Collapses to a 1-Complex Is NP-Complete". In: *Discrete Geometry for Computer Imagery*. Ed. by David Coeurjolly et al. Berlin, Heidelberg: Springer, 2008, pp. 177–188. ISBN: 978-3-540-79126-3. DOI: 10.1007/978-3-540-79126-3_17.

[67] C. R. F. Maunder. *Algebraic Topology*. Courier Corporation, Jan. 1, 1996. 414 pp. ISBN: 978-0-486-69131-2. Google Books: YkyizIcJdK0C.

[68] Wes McKinney. "Data Structures for Statistical Computing in Python". In: Python in Science Conference. Austin, Texas, 2010, pp. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

[69] Frank D. McSherry. "Privacy Integrated Queries: An Extensible Platform for Privacy-Preserving Data Analysis". In: *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. SIGMOD '09. New York, NY, USA: Association for Computing Machinery, June 29, 2009, pp. 19–30. ISBN: 978-1-60558-551-2. DOI: 10.1145/1559845.1559850.

[70] Hourie Mehrabiun and Behnaz Omoomi. "Personalized Privacy Preserving Method for Social Networks Graph K-Anonymization". In: *Computer and Knowledge Engineering* 6.1 (Apr. 1, 2023), pp. 37–46. ISSN: 2538-5453. DOI: 10.22067/cke.2023.63240.0.

[71] James R. Munkres. *Elements Of Algebraic Topology*. 1st ed. CRC Press, Mar. 5, 2018. 465 pp. ISBN: 978-0-429-96246-2. DOI: 10.1201/9780429493911.

[72] Arvind Narayanan and Vitaly Shmatikov. "Robust De-anonymization of Large Sparse Datasets". In: *2008 IEEE Symposium on Security and Privacy (Sp 2008)*. 2008 IEEE Symposium on Security and Privacy (Sp 2008). May 2008, pp. 111–125. DOI: 10.1109/SP.2008.33.

[73] Mehmet Ercan Nergiz, Maurizio Atzori, and Chris Clifton. "Hiding the Presence of Individuals from Shared Databases". In: *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*. SIGMOD '07. New York, NY, USA: Association for Computing Machinery, June 11, 2007, pp. 665–676. ISBN: 978-1-59593-686-8. DOI: 10.1145/1247480.1247554.

[74] Hiep H. Nguyen, Abdessamad Imine, and Michaël Rusinowitch. "Anonymizing Social Graphs via Uncertainty Semantics". In: *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. ASIA CCS '15. New York, NY, USA: Association for Computing Machinery, Apr. 14, 2015, pp. 495–506. ISBN: 978-1-4503-3245-3. DOI: 10.1145/2714576.2714584.

[75] Adrián Tobar Nicolau, Javier Parra-Arnau, and Jordi Forné. "A Taxonomy of Syntactic Privacy Notions for Continuous Data Publishing". In: *IEEE Access* 12 (2024), pp. 38490–38511. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2024.3368852.

[76] Giovanni Paolini. "Collapsibility to a Subcomplex of a Given Dimension Is NP-Complete". In: *Discrete & Computational Geometry* 59.1 (Jan. 1, 2018), pp. 246–251. ISSN: 1432-0444. DOI: 10.1007/s00454-017-9915-6.

[77] Jordi Soria-Comas and Josep Domingo-Ferrer. "Big Data Privacy: Challenges to Privacy Principles and Models". In: *Data Science and Engineering* 1.1 (Mar. 1, 2016), pp. 21–28. ISSN: 2364-1541. DOI: 10.1007/s41019-015-0001-x.

[78] Klara Stokes and Vicenç Torra. "MULTIPLE RELEASES OF K-ANONYMOUS DATA SETS AND k-ANONYMOUS RELATIONAL DATABASES". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 20.06 (Dec. 2012), pp. 839–853. ISSN: 0218-4885. DOI: 10.1142/S0218488512400260.

[79] Latanya Sweeney. "K-ANONYMITY: A MODEL FOR PROTECTING PRIVACY". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 10.05 (Oct. 2002), pp. 557–570. ISSN: 0218-4885. DOI: 10.1142/S0218488502001648.

[80] Chih-Hua Tai et al. "Privacy-Preserving Social Network Publication against Friendship Attacks". In: *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '11. New York, NY, USA: Association for Computing Machinery, Aug. 21, 2011, pp. 1262–1270. ISBN: 978-1-4503-0813-7. DOI: 10.1145/2020408.2020599.

[81] Martin Tancer. "Recognition of Collapsible Complexes Is NP-Complete". In: *Discrete & Computational Geometry* 55.1 (Jan. 1, 2016), pp. 21–38. ISSN: 1432-0444. DOI: 10.1007/s00454-015-9747-1.

[82] The Matplotlib Development Team. *Matplotlib: Visualization with Python*. Version v3.9.2. Zenodo, Aug. 13, 2024. DOI: 10.5281/ZENODO.592536.

[83] The pandas development team. *Pandas-Dev/Pandas: Pandas*. Version v2.2.2. Zenodo, Apr. 10, 2024. DOI: 10.5281/zenodo.10957263.

[84] 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. New York, NY, USA: Association for Computing Machinery, Mar. 10, 2009, pp. 218–227. ISBN: 978-1-60558-394-5. DOI: 10.1145/1533057.1533088.

[85] Svetlana Ulianova. *Cardiovascular Disease Dataset*. 2019. URL: https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset (visited on 08/27/2024).

[86] Pauli Virtanen et al. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17.3 (Mar. 2, 2020), pp. 261–272. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-019-0686-2.

[87] Jincheng Wang et al. "Topology-Theoretic Approach to Address Attribute Linkage Attacks in Differential Privacy". In: *Computers & Security* 113 (Feb. 1, 2022), p. 102552. ISSN: 0167-4048. DOI: 10.1016/j.cose.2021.102552.

[88] Ke Wang and Benjamin C. M. Fung. "Anonymizing Sequential Releases". In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '06. New York, NY, USA: Association for Computing Machinery, Aug. 20, 2006, pp. 414–423. ISBN: 978-1-59593-339-3. DOI: 10.1145/1150402.1150449.

[89]   J. H. C. Whitehead. "Simplicial Spaces, Nuclei and m-Groups". In: *Proceedings of the London Mathematical Society* s2-45.1 (1939), pp. 243–327. ISSN: 1460-244X. DOI: `10.1112/plms/s2-45.1.243`.

[90]   Leon Willenborg and Ton de Waal. *Elements of Statistical Disclosure Control.* Springer Science & Business Media, Dec. 6, 2012. 273 pp. ISBN: 978-1-4613-0121-9. Google Books: `eVnSBwAAQBAJ`.

[91]   Royce J. Wilson et al. *Differentially Private SQL with Bounded User Contribution.* Nov. 25, 2019. DOI: `10.48550/arXiv.1909.01917`. arXiv: `1909.01917 [cs]`. Pre-published.

[92]   Wentao Wu et al. "K-Symmetry Model for Identity Anonymization in Social Networks". In: *Proceedings of the 13th International Conference on Extending Database Technology.* EDBT '10. New York, NY, USA: Association for Computing Machinery, Mar. 22, 2010, pp. 111–122. ISBN: 978-1-60558-945-9. DOI: `10.1145/1739041.1739058`.

[93]   Xintao Wu et al. "A Survey of Privacy-Preservation of Graphs and Social Networks". In: *Managing and Mining Graph Data.* Ed. by Charu C. Aggarwal and Haixun Wang. Advances in Database Systems. Boston, MA: Springer US, 2010, pp. 421–453. ISBN: 978-1-4419-6045-0. DOI: `10.1007/978-1-4419-6045-0_14`.

[94]   Xiaokui Xiao, Guozhang Wang, and Johannes Gehrke. "Differential Privacy via Wavelet Transforms". In: *IEEE Transactions on Knowledge and Data Engineering* 23.8 (Aug. 2011), pp. 1200–1214. ISSN: 1041-4347. DOI: `10.1109/TKDE.2010.247`.

[95]   Bin Yang, Issei Sato, and Hiroshi Nakagawa. "Bayesian Differential Privacy on Correlated Data". In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data.* SIGMOD '15. New York, NY, USA: Association for Computing Machinery, May 27, 2015, pp. 747–762. ISBN: 978-1-4503-2758-9. DOI: `10.1145/2723372.2747643`.

[96]   Xiaowei Ying and Xintao Wu. "On Link Privacy in Randomizing Social Networks". In: *Knowledge and Information Systems* 28.3 (Sept. 1, 2011), pp. 645–663. ISSN: 0219-3116. DOI: `10.1007/s10115-010-0353-5`.

[97]   Elena Zheleva and Lise Getoor. "Preserving the Privacy of Sensitive Relationships in Graph Data". In: *Privacy, Security, and Trust in KDD.* Ed. by Francesco Bonchi et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 153–171. ISBN: 978-3-540-78478-4. DOI: `10.1007/978-3-540-78478-4_9`.

[98]   Bin Zhou and Jian Pei. "Preserving Privacy in Social Networks Against Neighborhood Attacks". In: *2008 IEEE 24th International Conference on Data Engineering.* 2008 IEEE 24th International Conference on Data Engineering. Apr. 2008, pp. 506–515. DOI: `10.1109/ICDE.2008.4497459`.

[99]   Lei Zou, Lei Chen, and M. Tamer Özsu. "K-Automorphism: A General Framework for Privacy Preserving Network Publication". In: *Proceedings of the VLDB Endowment* 2.1 (Aug. 1, 2009), pp. 946–957. ISSN: 2150-8097. DOI: `10.14778/1687627.1687734`.

A. Syntactic Graph Data Privacy Surveys Summarized

In the following we summarize, contrast and comment on the classifications of graph data privacy "methods" given by surveys [93, 1, 53]. In Table A.1 we show the overlap between the given classifications, and notions whose mechanisms were listed in the corresponding categories.

Classifications of [93]:

- k-anonymity via edge modification: Involves modifying edges so that each node is indistinguishable with at least $k-1$ other nodes in terms of some structural patterns.

- Edge-randomization techniques are resilient to structural attacks according to [93].
  - These methods involve randomly adding or deleting k edges, or randomly switching existing edges
  - A potential flaw of these methods: Real world data is usually highly correlated in a low dimensional space while randomly added noise is distributed equally over all dimensions [93].
  - Feature Preserving Randomization: Consideration of directly generating synthetic graphs given a set of features.
  - Spectrum Preserving Randomization.
  - Markov Chain based Feature Preserving Randomization.
  - None of these methods come with a formal notion which they fulfill, which makes them inadequate for this comparison, despite their potential benefits.

- Generalization-based generalization (Including clustering)
  - Problem of generalization: Released network only contains summary of structural information about the original graph, so users have to generate some random sample instances of released graph and thus uncertainty might arise in later analysis since the samples came from numerous possible worlds [93].

The classification [1] provides is in perturbation and non-perturbation methods, where in non-perturbation methods vertices and edges might be suppressed and generalized but never altered, whereas the perturbation methods additionally inject noise to the output. However, we skip the random graph editing models, since these do not define a notion. The survey also adds a category for DP-based mechanisms. We will not address these here either, as we address them in section 5.2. Their remaining classification is structured as follows:

- Non-Perturbation Privacy Preservation Methods
  - Random Graph Editing Models

- *k*-Anonymization Approaches
  * Degree anonymization
  * Neighborhood anonymization
  * Structure-free anonymity techniques
- Clustering Based
- Probabilistic Privacy Preservation Models

The survey [1] observes some points that are also good to point out in this context:

- Arbitrary modification of edges or vertices is does not necessarily create privacy for all users, and random perturbation mechanisms cannot attain a significant degree of anonymity without diluting graph features.

- Eigenvalues of the adjacency matrix and of the Laplacian matrix should be preserved to maintain utility.

- "Randomization techniques are unique as they are not focused on addessing special adversarial attack"

- They also quote [59], who show that randomization techniques *may* achieve meaningful levels of anonymization while preserving graph characteristics.

For the *k*-anonymity notions Abawajy, Ninggal, and Herawan note that they have the goal to mitigate vertex re-identification and that their main difference is in the (assumed) prior knowledge of the attacker. However for the structure-free (*k*-)anonymity techniques they note that these models exploit the inherent symmetry in graphs. The clustering based *k*-anonymity approaches aim to partition the edges and vertices into a minimum of super clusters, while maintaining structural information, however the survey writes that finding a globally optimal partition is difficult, and that for a legitimate analysis, a graph has to be sampled from all likely clusters, which leads to sampling errors.

For DP they mention, that it would lead to unacceptably low data quality in certain data analysis situations. They also note that one of the methods they showcase is not publishing the graph (but rather only estimates of degree distributions), and that in some case the structural properties are of great interest, so that publishing the anonymized graph would be preferable.

Review [53] also has a tree based taxonomy, on the first level being graph-modification techniques and differential privacy. The reasoning for this distinction is that according to [53] DP only works on queries. This conflicts with [60] who also mentions DP graph publishing methods. We also point out that this is not a taxonomy of protections given, but rather of in which ways the graph is modified. The review does not contain any comparison of notions.

Within graph modification techniques the distinctions [53] makes are between the following categories:

- Edge and Vertex modification techniques
  - Random perturbation
    * "In this method, privacy is not guaranteed but a probabilistic-identification model." Which means that this is no notion we could compare guarantees with.

* They note about one of the methods that, it creates "randomizaiton techniques that preserve the features of the graph effectively without giving them a lot of privacy", however "privacy" is not qualified or quantified.

* Similarly they note about a "proposed entropy-based quantification" based on posterior belief probabilities "the latest research of randomization methods has shown that they reach a significant anonymization degree and retain much of the original graph characteristics"; again without any formal quantification.

– Constrained perturbation — this includes k-Anonymity methods (notions)

* The authors mention that the privacy guarantee depends on the opponents knowledge (specified via quasi identifier and vertex degree).

* One of the mentioned papers [97]"Proposed the inferring sensitive relationship problem from the anonymized structure as link identification." which is close to our approach of using Association Privacy to protect the inferred sensitive relationships.

* All of the methods mentioned in this category "have certain drawbacks that lead to information loss and there is no technique that assures privacy in all aspects of node security, edge security".

• Generalization and Clustering

– "this method is not suitable for analyzing the local structures of the graph but a good method for aggressive queries answering"

– The survey states, that "generalization methods have a strong degree of privacy although they complicate the study of local measures and metrics", but again does not give a definition of what degree of privacy means here, or comparison with other

Class. of [93]	Classifications of [1]			Classifications of [53]		Notion
k-Anonymity via Edge Modification	Non-Pertubation Privacy Preservation	k-Anonymity	Degree anonymization	Edge/ Vertex modification techniques	Constrained Perturbation	k-degree-anonymity/ k-anonymous degree sequence [62] k^2-degree anonymity [80] k-degree-subset-anonymity [19]
			Neighborhood anonymization			k-neighborhood anonymous [98] 1-hop anonymity [84] k-candidate anonymity [45]
			Structure-free anonymity			k-automorphic graph [99] k-symmetry [92] k-isomorphic graph [18]
Edge Randomization		Probabilistic Privacy Preservation Models			Random perturbation/ Obfuscation	k-obfuscation [11]
Generalization-based		Clustering-Based		Generalization/ Clustering		k-anonymity clustered [45]

Table A.1.: Classifications of graph privacy "methods" given by surveys [93, 1, 53]