D2P*t*: Privacy-Aware Multiparty Data Publication

J. H. Nielsen, D. Janusz, J. Taeschner, J.-C. Freytag

Database and Information Systems Group
Humboldt-Universität zu Berlin
{nielseja,janusz,taeschnj,freytag}@informatik.hu-berlin.de

Abstract: Today, publication of medical data faces high legal barriers. On the one hand, publishing medical data is important for medical research. On the other hand, it is neccessary to protect peoples' privacy by ensuring that the relationship between individuals and their related medical data remains unknown to third parties. Various data anonymization techniques remove as little identifying information as possible to maintain a high data utility while satisfying the strict demands of privacy laws.

Current research in this area proposes a multitude of concepts for data anonymization. The concept of k-anonymity allows data publication by hiding identifying information without losing its semantics. Based on k-anonymity, the concept of t-closeness incorporates semantic relationships between personal data values, therefore increasing the strength of the anonymization. However, these concepts are restricted to a centralized data source.

In this paper, we extend existing data privacy mechanisms to enable joint data publication among multiple participating institutions. In particular, we adapt the concept of t-closeness for distributed data anonymization. We introduce *Distributed two-Party t-closeness (D2Pt)*, a protocol that utilizes cryptographic algorithms to avoid a central component when anonymizing data adhering the t-closeness property. That is, without a trusted third party, we achieve a data privacy based on the notion of t-closeness.

1 Introduction

Over the past years, various incidents of *privacy breaches* have fueled the growing demand for preserving privacy of individuals. The ever increasing digital footprint of individuals makes it easier to gather *sensitive private information*. Abuses in the past show that the collection of data sets imposes a risk for revealing sensitive private information [Swe97]. On the other hand, data collection and exchange is necessary to support the needs of institutions that rely on it, e. g., medical institutions conducting a clinical trial. Therefore, taking privacy protection approaches to an extreme by not releasing data at all is no solution. Only revealing non-sensitive information does not provide a feasible solution as well, since medical institutions rely on information about sensitive data, e. g., diseases. The challenge is to find a balance between the privacy interests of individuals and the interests of organizations and companies to gain access to personal data.

Research in the area of privacy protection has developed the domains of Privacy-Preserving Data Mining (PPDM) and Privacy-Preserving Data Publishing (PPDP) [CT13]. Both areas aim to allow access to data sets while preserving privacy of individuals. However, both areas differ in how to handle data sets. The focus of PPDM is to answer queries targeting disclosed data sets in a privacy preserving way. The focus of PPDP is to generate and publish a complete data set that preserves the privacy of individuals. This paper will focus on microdata protection techniques of PPDP.

One major field of application for PPDP are clinical trials. Clinical trials rely on the publication of private data. Especially, in the field of medicine data is considered highly sensitive. The implied trade-off becomes hard to solve. First, to maintain the utility of the data. Second, to keep the data private at the same time. Therefore, the essential idea when publishing clinical trial data is to achieve a minimal information loss without violating the privacy of individuals.

The Health Insurance Portability and Accountability Act (HIPAA) [US96] is a legislative effort to protect individuals when releasing information about them. For example, HIPAA specifically addresses the removal of personally identifiable information when publishing medical data of individuals (also known as the HIPAA Privacy Rule and the HIPAA Safe Harbor Provision). The notion of privacy and sensitive information are considered to be either to rigid or to imprecise to protect the privacy of individuals [LeF07].

To overcome the limitations imposed by the law, different concepts have been proposed for PPDP. The concept of *k-anonymity* seeks to protect sensitive private information by altering specific attributes using syntactical rules [Swe02b]. An enhancing notion of privacy is the concept of *t-closeness*, enriching the concept of k-anonymity with the semantic relationship between attribute values [LLV07]. However, both concepts operate on a single data set.

Generally, privacy concepts that consider sensitive attributes (SAs) are limited to a single sensitive attribute [FAN11]. In contrast, multi-center clinical trials carried out by more than one institution rely on the publication of data containing more than one SA. Each institution collects different data about the same patient, thus forming a *vertically partitioned* data collection. Such a scenario creates the *multiple sensitive attributes (MSA)* problem [FAN11], i. e., multiple sensitive attributes collected by multiple institutions. Several approaches recognize the importance of MSA without providing necessary algorithms tailored to the specific needs of a distributed data collection environment [DB12].

In the light of this discussion, the contribution of this paper is threefold: (*1*) We extend the concept of t-closeness to a distributed vertically partitioned environment with multiple sensitive attributes and analyze the demands imposed on the concept of t-closeness, (*2*) we develop a protocol named *D2Pt*, which achieves data anonymization utilizing the concept of t-closeness and (*3*) we present an experimental evaluation of our protocol D2P*t*. Furthermore, we compare D2P*t* to previous state-of-the-art protocols that do no utilize semantic data anonymization.

The remainder of this paper is organized as follows. In section 2, we provide background information on data anonymization techniques following the notion of PPDP. In section 3, we present our extended concept of t-closeness and introduce our protocol D2P*t*. In section 4, we show our evaluation results as well as an analysis of our results. In section 5, we provide an overview on related work. We conclude and outline future work in section 6.

2 Background

Nowadays, data publication focuses on publishing data as *microdata*, typically in tabular form [CDCdVFS07]. Microdata contains unaltered data statistics on individuals [Dal77]. An example of a microdata table can be found in Table 1a.

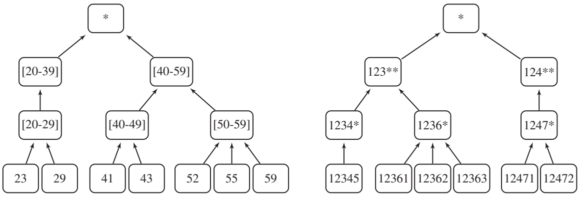
2.1 Centralized data-anonymization

Attributes contained in a microdata-table can be classified into three categories: *identifiers (IDs)*, *sensitive attributes (SAs)* and *quasi-identifiers (QIDs)*. The ID uniquely identifies an individual. An example is the full name of an individual. The SA specifies an attribute an individual does not want to be associated to, e. g., a disease. For that reason the ID is removed from a microdata publication. This process is called *de-identification* [Swe97]. In [Swe97] Sweeney showed that de-identification does not suffice to protect microdata publications. This is due to other publicly available data sets, like voter lists. Voter lists contain the name as well as demographic information like the date of birth, age and gender of an individual. The combination of these attributes values is often unique [Swe97], hence they are termed QIDs. The de-identified microdata table can be *linked* to the publicly available data by joining them on the QID.

k-anonymity To protect against linking, Sweeney introduced the concept of *k-anonymity* in [Swe02b]. Intuitively, k-anonymity alters the values of the QIDs, thus making every individual in that microdata indistinguishable from at least $k - 1$ individuals w. r. t. to QID values [Swe02b]. The values are not randomly altered, rather they are being removed or replaced by a "less specific but semantically consistent value" [Swe02a]. This mechanism is called *generalization and suppression* [FED94]. A group of individuals in a microdata table is said to form an *equivalence class (EC)* if they equal in their QID values.

A semantically consistent value is determined by the use of a *value generalization hierarchy (VGH)* [Sam01]. A VGH is a directed, balanced tree describing the proper generalization of attribute values. The specific values are located in the leaves of the tree. the more general values are located in the inner nodes. The root node consists of the most general value or the value '*', which indicates the suppression of a value. A VGH for the attribute AGE can be found in Figure 1a. An example for the generalization of the value 22 of attribute AGE is the age span $[20 - 29]$.

The Datafly-algorithm Meyerson et al. and Aggarwal et al. have shown that the problem of an optimal anonymization is NP-hard [MW04, AFK$^+$05]. Therefore, approximate solutions to achieve k-anonymity exist. In [Swe97] Sweeney suggested the *Datafly*-algorithm. In its basic version the Datafly-algorithm uses *full-domain generalization* without suppression, i. e., at every state of the algorithm each QID value is at the same level of the VGH. In every iteration of the algorithm, Datafly chooses one out of the QID attributes, specifically the attribute containing the most different values. The algorithm then



(a) VGH for QID AGE

(b) VGH for QID ZIP

Figure 1: Value generalization hierarchies

generalizes the values of that attribute, i. e., replacing every occurrence of a value with the value one level up in the VGH. It continues until every individual in the microdata table is indistinguishable from at least $k-1$ other individuals in that table w. r. t. QID values. This algorithm will serve as an example throughout this paper, therefore, we will elaborate more and show its operating principle using an example.

Example 1. Consider Table 1a to be a microdata table and the goal to create an anonymous version of that table satisfying 2-anonymity. Let attribute ID be the identifier, AGE and ZIP the QID with corresponding VGHs in Figure 1a and 1b. The SAs are given by the attributes DISEASE and TREATMENT. To provide an easy to follow example, we will not remove the ID. In the first iteration of Datafly the algorithm would pick attribute AGE for generalization, since it has the most different values: nine compared to eight of attribute ZIP. Every value is replaced by a more general value. Attribute ZIP still contains eight unique values, hence another generalization is performed on attribute ZIP. This table satisfies 1-anonymity, since the combination of ZIP and AGE is unique for tuple ID 5 (1236* and $[50-59]$. In order to satisfy 2-anonymity another iteration has to be performed. The generalization of attribute ZIP gives the resulting 2-anonymous table depicted in Table 1b, i. e., every individual in the table is indistinguishable from at least another individual in that table. The equivalence classes (ECs) are marked by dashed horizontal lines.

ID	ZIP	AGE	DISEASE	TREATMENT
1	12345	23	Gastric ulcer	Antacid
2	12345	29	Gastritis	Acid-reducing drug
3	12363	41	Flu	Antipyretic drug
4	12361	43	Stomach cancer	Cytostatic drug
5	12362	59	Pneumonia	Antibiotics
6	12471	52	Bronchitis	Antibiotics
7	12473	55	Flu	Antipyretic drug

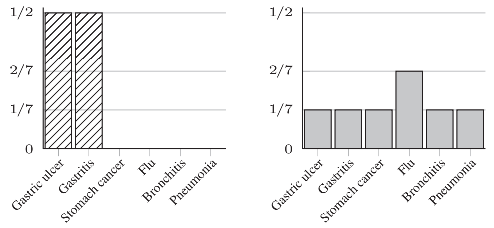
(a) Microdata-table

ID	ZIP	AGE	DISEASE	TREATMENT
1	123**	[20-29]	Gastric ulcer	Antacid
2	123**	[20-29]	Gastritis	Acid-reducing drug
3	123**	[40-49]	Flu	Antipyretic drug
4	123**	[40-49]	Stomach cancer	Cytostatic drug
5	123**	[50-59]	Pneumonia	Antibiotics
6	124**	[50-59]	Bronchitis	Antibiotics
7	124**	[50-59]	Flu	Antipyretic drug

(b) 2-anonymous table

Table 1: 2-anonymization of a microdata table

t-closeness The focus on the QIDs turned out to be a weak point of the concept of k-anonymity [MKGV07]. In [LLV07] Li et al. criticized the concept of k-anonymity for not taking the values of the SAs into account. Li et al. pointed out, that the semantic



(a) Frequency distribution of attribute DISEASE for the first EC (IDs {1,2})

(b) Frequency distribution of attribute DISEASE for the whole table

Figure 2: Comparison of frequency distribution

relationships between SA values of individuals may allow *attribute disclosure*. i. e., the association of an individual with the value of its SA or a *semantically close* value [LLV07]. To address the notion of closeness, Li et al. use the earth-movers distance (EMD) [RTG00].

The EMD is a metric used in image processing to measure the similarity of two pictures. The pictures are being described by the frequency distribution of colors occurring in the picture. Intuitively the EMD measures the amount of work necessary to transform one distribution into the other. The result is a value in the interval of $[0, 1]$. The closer the value is to 0 the closer the two distributions are.

Adapted to the use case of data anonymization, the similarity of attribute values is measured by comparing the frequency distribution of attribute values for every EC to the frequency distribution of the whole table. If the result is closer to 1, the values in an EC are very similar, thus allowing to infer information [LLV07]. Hence, a value closer to 0 is desirable. An example of the frequency distribution for the first EC (IDs {1,2}) for attribute DISEASE of Table 1b to the whole Table 1b is depicted in Figure 2a and Figure 2b. Based on Table 1b and a simplified VGH of attribute DISEASE in [LLV07] the table satisfies 0.57-closeness w. r. t. DISEASE[1]. If privacy requirements demand a lower t-closeness value, the table could be generalized once again using the Datafly-algorithm. The resulting table is a maximally generalized table containing only values of the root nodes of the VGHs. It satisfies 0-closeness, since it consists of only one EC.

2.2 Distributed data-anonymization

As outlined in section 1, today's interconnected society places new demands on data privacy. The challenge is to adapt to a computing environment where data is distributed across multiple distinct sites. Throughout this paper, we focus on physically distributed data *partitioned* across two distinct sites, i. e., medical institutions. In such an environ-

[1] Details of this calculation are left out due to space considerations. The reader is referred to [LLV07] for an in depth explanation of t-closeness

ment, data can be vertically or horizontally partitioned. In the following we focus on vertically partitioned data, i. e., data about the same individuals stored at multiple sites.

One problem when jointly computing an anonymization on data split across multiple sites, is to define a *security model*. A security model describes demands imposed on participating sites and what can be learned from executing the *protocol*, i. e., the distributed data anonymization algorithm. A typical security model is that of an *honest-but-curious (HBC) adversary*, i. e., no site gains information about another sites input other than can be inferred from the own input and the result of the computation [Gol04]. In an HBC model, each site acts accordingly to the protocol, i. e., every site is truthful about its input and further steps of the protocol [Gol04]. We adhere to this security model for the remainder of this paper. Solutions to this model involve either the use of a *trusted third party (TTP)* or the utilization of a *secure multiparty computation (SMC)* protocol [Gol04].

Using a TTP may not be possible at all times. Therefore, the paradigm of SMC describes a set of protocols that try to simulate the use of a TTP [Gol04]. This is achieved by using cryptographic functions hiding the input of participating sites and revealing nothing but the result of the computation.

One way of protecting the input is the use of a *commutative encryption scheme* as described by Shamir et al. in [SuLMA80]. An encryption scheme uses a *cryptographic function* $E_k : M \to M$, that transforms a *plaintext* $m \in M$ into a *cyphertext* $m \in M$ using a *key* k. We omit the details of E. The important property of a commutative encryption scheme is the independence of the order of encryption, i. e., given the encryption function E_k, two keys $0, 1$ and a plaintext $m \in M$, the following commutativity property holds: $E_0 \left(E_1 \left(m \right) \right) = E_1 \left(E_0 \left(m \right) \right)$.

2.3 DPP$_2$GA: distributed k-anonymity

The *Distributed Privacy-Preserving two-Party Generic Anonymizer (DPP$_2$GA)* protocol by Jiang et al. [JC05] implements distributed k-anonymity on vertically partitioned data. Its goal is to create a *globally* k-anonymous data set from two *locally* k-anonymous data sets, stored at different sites. During execution of the protocol, k-anonymity of the locally stored data sets is preserved. This definition of distributed privacy is related to the definition of SMC, but differs in that it allows *some* degree of information gain.

Before discussing the protocol in more detail we will outline its key steps: (*1*) Ensure local k-anonymity, (*2*) exchange encrypted information about local ECs, (*3*) ensure global k-anonymity and (*4*) join local data sets on a global identifier. We will elaborate on the key steps by using an example.

Example 2. The prerequisite of the protocol is a vertically partitioned table, distributed across two sites. An example is given in Table 2a reusing the running example from Table 1a split into two tables. For simplicity, we omit values for the SAs. The subscript number close to the attribute name indicates that the attribute is either stored at site 0 or site 1. As before, dashed horizontal lines indicate ECs. Following this abstract example, each local table is 1-anonymous, i. e., at least one individual is unique w. r. t. its QID values. We

ID_0	ZIP_0
1	12345
2	12345
3	12363
4	12361
5	12362
6	12471
7	12473

ID_1	AGE_1
1	23
2	29
3	41
4	43
5	59
6	52
7	55

(a) Initial state

ID_0	ZIP_0
1	1234*
2	1234*
3	1236*
4	1236*
5	1236*
6	1247*
7	1247*

ID_1	AGE_1
1	[20-29]
2	[20-29]
3	[40-49]
4	[40-49]
5	[50-59]
6	[50-59]
7	[50-59]

(b) 1st iteration

ID_0	ZIP_0
1	123**
2	123**
3	123**
4	123**
5	123**
6	124**
7	124**

ID_1	AGE_1
1	[20-39]
2	[20-39]
3	[40-59]
4	[40-59]
5	[40-59]
6	[40-59]
7	[40-59]

(c) 2nd iteration

Table 2: Distributed 2-anonymity using DPP_2GA

will assume that the participating sites agreed on publishing a globally 2-anonymous table. Therefore, following step (*1*), every site first creates a locally 2-anonymous version of its data set. As Jiang et al. [JC05] noted, this can be achieved using any k-anonymization algorithm. For simplicity, we assume the use of the Datafly-algorithm.

After the first iteration of Datafly on the local data sets, we obtain different ECs on both sites. The global result is identical to the result outlined in example 1. Hence, we observe the ECs depicted in Table 2b. Next, we investigate step (*2*) of the protocol. This step can be divided into five parts: (*2.1*) send an encrypted version of the ECs to the other site, (*2.2*) receive an encrypted version of the ECs from the other site, (*2.3*) encrypt received EC with own key and send it back to the other site, (*2.4*) receive an encrypted version of the EC sent to the other site in part (*2.1*) and (*2.5*) compare the encrypted multisets. To satisfy the k-anonymity requirement the sites need to exchange information about the equivalent classes in an encrypted, thus privacy-preserving way. Using the commutative encryption function, introduced in subsection 2.2, the two sites exchange their multisets.

We will explain this step by continuing our previous example, focusing on the first EC. Each site is now in the possession of the following encrypted multisets: site $0 : \{\{E_0(1), E_0(2)\}, ...\}$ and site $1 : \{\{E_1(1), E_1(2)\}, ...\}$. These encrypted IDs are being sent to the other site (part (*2.1*)). Site 0 now owns an encrypted version of the ECs of site 1. Furthermore, site 1 now owns an encrypted version of the ECs of site 0 (part (*2.2*)). Due to the encryption, neither site is able to read the IDs, therefore the k-anonymity requirement is still intact. As outlined in part (*2.3*), every site encrypts the version just received by the other site, with its own key and sends it back to the corresponding site. Now, every site owns two versions of the encrypted ECs: First, one originally encrypted by themselves and returned encrypted yet again by the other site. Second, the version originally received from the other site and now encrypted by themselves (part (*2.4*)). Since, the commutativity property holds for the encryption function E, we are now able to compare the ECs without knowing the actual values of the IDs (part (*2.5*)). For that purpose, Jiang et al. proved, that a local k-anonymization is globally k-anonymous, under the following circumstances: (*1*) For every encrypted ID in each EC in one multiset, there exists another EC in the other multiset containing the same ID. (*2*) The cut of those two sets is of cardinality at least k. We will refer to this property as the *multiset-equality property*. Continuing our example, this is not the case for ID 5 as can be seen in Table 2b. Hence, the local k-anonymizations are not globally k-anonymous. Therefore, another iteration of the Datafly-algorithm has to be performed on the local data. Afterwards, the comparison has to be repeated. This process continues until a valid global k-anonymization is found. An example for a valid global k-anonymization can be seen in Table 2c.

Finally, in step (*4*) of the protocol the k-anonymous partitions have to be joined on the ID in order to form a k-anonymous publication. This can be achieved by using a secure join, as proposed by Jiang et al. in [JC06]. We will omit an explanation of this part, referring to [JC06] for further details.

3 Distributed two-party t-closeness

In this section, we present our protocol D2P*t*, which extends the DPP$_2$GA data anonymization technique by the concept of t-closeness.

3.1 Preliminary assumptions

For the following considerations, we make the same assumptions as in subsection 2.3, i. e., the data is vertically partitioned across two independent sites. Furthermore, the sites agreed on the parameters of the anonymization protocol prior to executing the protocol, e. g., k-value for k-anonymity, t-value for t-closeness, a common ID describing the same individual on both sites, as well as the parameters used for the commutative encryption scheme. In addition we use the same security model used in subsection 2.3, e. g., an HBC adversary. Lastly, we assume the privacy model described in [JC05], thus allowing some inference of information, hence weakening the rules of SMC. Since protocols of SMC are costly and add an additional layer of complexity, we decided to lose the absolute security of SMC but gain an intuitive and easy to follow protocol. Also, methods of computer and communication security are out of scope of this work. As Sweeney emphasizes in [Swe02b], we state that every entry in the table be specific to one individual only.

3.2 Problem statement

To clarify our concept of t-closeness in a distributed environment, we have to discuss some important observations. As we have shown in subsection 2.3, the concept of k-anonymity can be transferred to a distributed environment. Enhancing this approach with the concept of t-closeness has two shortcomings. First, as was shown in section 2.1, the t-closeness property incorporates the notion of semantics of SA values to an anonymization. Since the SA is the attribute an anonymization is trying to protect, information about it cannot be exchanged directly. Therefore, only relying on the cut-operation as an indicator of a globally valid anonymization is not enough. The cut may shrink an EC as long as its cardinality is still greater or equal to k. A reduced EC has a drastic influence on the t-closeness property, thus shrinking an EC is not an option when using t-closeness. Secondly, a joined data publication imposes several problems. On the one hand side, it is

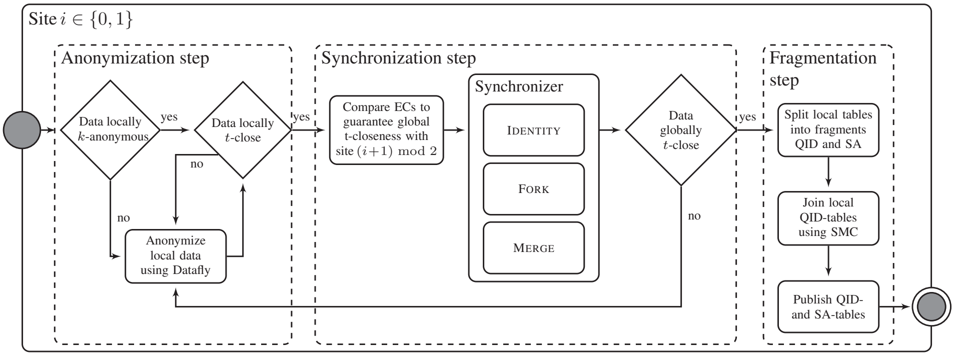

Site $i \in \{0,1\}$
Anonymization step
Data locally k-anonymous
yes
Data locally t-close
no
no
Anonymize local data using Datafly
yes
Synchronization step
Compare ECs to guarantee global t-closeness with site $(i+1) \bmod 2$
Synchronizer
IDENTITY
FORK
MERGE
Data globally t-close
no
yes
Fragmentation step
Split local tables into fragments QID and SA
Join local QID-tables using SMC
Publish QID- and SA-tables

Figure 3: Flowchart of the protocol D2P*t*

desirable to publish the combination of every SA for each individual. On the other hand, as Fang et al. [FAN11] pointed out, this changes the characteristics of the SA. Since the original values of each SA are known to the site publishing it, this site could re-identify the value in the anonymized data set, thus re-identifying an individual. Therefore, the SA values transform into additional QIDs, which we term *identifying sensitive attributes (iSAs)*. iSAs introduce a new threat to k-anonymity. Fang et al. [FAN11] identified this threat as the *background-join attack*. Lastly, it remains unclear how data privacy models like t-closeness, that operate on frequency distributions of SA values handle the case of multiple sensitive attributes (MSA). Computing t-closeness on the joint distribution of the SAs does not provide a feasible solution, since it will most likely be evenly distributed, thus providing no useful information.

Our observations lead to the following requirements for a global t-closeness in a distributed two-party environment having multiple sensitive attributes: (*1*) The released data must be t-close for every SA. (*2*) The k-anonymity-property must respect the background-join attack based on iSAs.

3.3 Overview

A schematic overview of our protocol D2P*t* can be found in Figure 3. It depicts an instantiation of the key components at one site. The protocol consists of three components: (*1*) the *Anonymizer*, (*2*) the *Synchronizer* and (*3*) the *Fragmenter*.

3.4 Anonymization phase: k-anonymity and t-closeness

The Anonymizer creates a locally k-anonymous data set, using an anonymization algorithm. For the sake of simplicity, we chose the Datafly-algorithm which was introduced in section 2.1. Although we are not limited to the Datafly-algorithm, as any algorithm

ID$_0$	ZIP$_0$
1	1234*
2	1234*
3	1236*
4	1236*
5	1236*
6	1247*
7	1247*

ID$_1$	AGE$_1$
1	[20-29]
2	[20-29]
3	[40-49]
4	[40-49]
5	[50-59]
6	[50-59]
7	[50-59]

(a) 1st iteration

ID$_0$	ZIP$_0$
1	123**
2	123**
3	123**
4	123**
5	123**
6	124**
7	124**

ID$_1$	AGE$_1$
1	[20-39]
2	[20-39]
3	[40-59]
4	[40-59]
5	[40-59]
6	[40-59]
7	[40-59]

(b) 2nd iteration

ID$_0$	ZIP$_0$
1	*
2	*
3	*
4	*
5	*
6	*
7	*

ID$_1$	AGE$_1$
1	*
2	*
3	*
4	*
5	*
6	*
7	*

(c) 3rd iteration

Table 3: Distributed t-closeness using D2Pt-IDENTITY

guaranteeing full-domain generalization is applicable to provide the anonymization phase. The k-anonymous data is handed over to the t-closeness-verifier, that in turn verifies the t-closeness of the data for a given t. If that check fails, the data is being handed back to the Anonymizer to produce a more general version of the anonymized table. After a finite number of iterations the table satisfies t-closeness. A table will satisfy t-closeness after a finite number of runs, since any generalization algorithm will produce a maximally generalized table at some point of its execution. The maximally generalized table is equal to the whole table thus satisfying 0-closeness, as outlined in section 2.1.

3.5 Synchronizer

The Synchronizer component handles the comparison of the two local anonymizations and decides whether they fulfill the needs of a globally t-close data set or not. It uses the mechanism of commutative encryption outlined in subsection 2.2 and subsection 2.3 to compare the anonymized tables produced by the two sites. In case of a negative result the local anonymization is handed back to the Anonymizer to perform another iteration of the anonymization algorithm. We developed three versions of the Synchronizer. Each one providing special properties to the protocol, as will be outlined below.

Synchronizer: IDENTITY The IDENTITY Synchronizer is the simplest form of a Synchronizer we developed. Its mode of operation is a straight-forward adaption of the synchronization protocol used by Jiang et al. [JC05]. Instead of demanding a cut of cardinality of at least k, each EC has to be identical on both sites.

Example 3. Revisiting our running example, Table 3a displays the state of the Synchronizer after the first iteration of Datafly. Again, we omit the SA values for clarity of the example. The dashed horizontal lines depict ECs. Due to ID 5 no equal ECs exist, thus the data is being handed back to the Anonymizer to create a more general anonymization. The result of the next synchronization attempt is shown in Table 3b. Once again the ECs do not match, hence demanding another generalization. This step is the final anonymization, since its result is a maximally generalized table, as can be seen in Table 3c. Therefore, the synchronization process is complete and the data is being transferred to the Fragmenter.

ID$_0$	ZIP$_0$	ID$_1$	AGE$_1$
1	1234*	1	[20-29]
2	1234*	2	[20-29]
3	1236*	3	[40-49]
4	1236*	4	[40-49]
5	1236*	5	[50-59]
6	1247*	6	[50-59]
7	1247*	7	[50-59]

(a) 1st iteration

ID$_0$	ZIP$_0$	ID$_1$	AGE$_1$
1*	1234*	1*	[20-29]
2*	1234*	2*	[20-29]
3	123**	3	[40-59]
4	123**	4	[40-59]
5	123**	5	[40-59]
6	124**	6	[40-59]
7	124**	7	[40-59]

(b) 2nd iteration

ID$_0$	ZIP$_0$	ID$_1$	AGE$_1$
1*	1234*	1*	[20-29]
2*	1234*	2*	[20-29]
3	*	3	*
4	*	4	*
5	*	5	*
6	*	6	*
7	*	7	*

(c) 3rd iteration

Table 4: Distributed t-closeness using D2Pt-FORK

Synchronizer: FORK The FORK Synchronizer is an improvement of the IDENTITY Synchronizer. We observed that, even though identical ECs exist, they are lost during the next generalization phase. The essential part of the FORK Synchronizer is its ability to mark identical ECs and exclude them from the next generalization phase.

This marking is also conducted in a privacy-preserving way. Therefore, every EC is assigned a separate ID, the *ec-ID*, which is also being encrypted using a commutative encryption scheme. After recognizing an identical EC, the FORK Synchronizer decrypts its ec-ID and sends it to the other site. In turn it receives an decrypted version of the ec-ID of the other site. Thus allowing both sites to decrypt the ec-ID and gain access to the information what EC to exclude from the next generalization phase.

Example 4. As an example review Table 4a derived from the initial Table 1a. The first EC containing the IDs 1 and 2 is equal on both sites. Thus the FORK-Synchronizer marks them for exclusion, depicted by an asterisk after the ID. The next iteration excludes the tuples from the first EC. The result is shown in Table 4b. Note, that no additional matching ECs are being found. Hence, a third iteration yields the result depicted in Table 4c.

Synchronizer: MERGE Revisiting the previous example, the following observation can be made. The IDs $\{3, 4, 5, 6, 7\}$ seem to be candidates for alignment, as can be seen in Table 5a. Unfortunately, the next generalization phase creates different ECs which render such a generalization impossible. The MERGE Synchronizer addresses this issue. We explain its behavior given the following example.

Example 5. The MERGE Synchronizer randomly selects one EC and adds its IDs to a queue Q which cannot contain duplicates. Let $\{6, 7\}$ be that EC, hence the queue consists of $Q = (6, 7)$. For each element in the queue on both sites the EC containing this element is being identified. The EC is then marked for not being available anymore, analogously to the FORK Synchronizer. We will select element 6. Every element in that EC is added to the queue, therefore, the elements $(5, 6, 7)$ from site 1 are added to the queue, e. g., $Q = (5, 6, 7)$. In the next step MERGE selects ID 5, leading to the addition of elements $(3, 4, 5)$ from site 0. The resulting queue is $Q = (3, 4, 5, 6, 7)$. After that no other elements are being added to the queue, indicating that MERGE found a valid merge of ECs, which, when generalized accordingly, will form a valid anonymization.

ID$_0$	ZIP$_0$
1	1234*
2	1234*
3	1236*
4	1236*
5	1236*
6	1247*
7	1247*

ID$_1$	AGE$_1$
1	[20-29]
2	[20-29]
3	[40-49]
4	[40-49]
5	[50-59]
6	[50-59]
7	[50-59]

(a) 1st iteration

ID$_0$	ZIP$_0$
1	1234*
2	1234*
3	*
4	*
5	*
6	*
7	*

ID$_1$	AGE$_1$
1	[20-29]
2	[20-29]
3	[40-59]
4	[40-59]
5	[40-59]
6	[40-59]
7	[40-59]

(b) 2nd iteration

GID	ZIP$_0$	AGE$_1$
1	1234*	[20-29]
1	1234*	[20-29]
2	*	[40-59]
2	*	[40-59]
2	*	[40-59]
2	*	[40-59]
2	*	[40-59]

GID	DISEASE
1	Gastric ulcer
1	Gastritis
2	Flu
2	Stomach cancer
2	Pneumonia
2	Bronchitis
2	Flu

GID	TREATMENT
1	Antacid
1	Acid-reducing drug
2	Antipyretic drug
2	Cytostatic drug
2	Antibiotics
2	Antibiotics
2	Antipyretic drug

(c) Resulting data publication

Table 5: Distributed t-closeness using D2Pt-MERGE

3.6 Fragmenter

Finally, the Fragmenter component is responsible for joining and releasing the data set. As its name suggests the resulting tables are not being joined and published as a whole. Rather, the data handling is twofold. On the one hand side, the quasi identifiers of sites 0 and 1 are being joined on the ID using a secure join protocol proposed by Jiang et al. [JC06]. On the other hand, the data is being partitioned, i. e., QIDs and SAs are being split into different tables. To maintain a reference from the QIDs to the SAs a new attribute, termed *global identifier (GID)* is inserted. The GID associates one SA value to at least k QID values, depending on the size of the EC. After the fragmentation the QIDs can be restored to their original values, because no direct association between a QID value and a SA value exists [XT06]. This process eliminates the risk for a background-join attack based on iSAs and, allow the t-closeness to be applied on each SA individually.

3.7 Privacy analysis

Privacy preservation We analyze our distributed anonymization protocol D2Pt w. r. t. compliance to k-anonymity and t-closeness. We show the validity of our approach using the following theorems with sketches of proofs.

The proof is split into three parts. First, we show compliance with k-anonymity during the execution of the protocol. Second, we extend this property to the t-closeness property. Lastly, we show that the fragmented result set adheres to the k-anonymity and t-closeness property.

Theorem 1. *The Synchronizer component introduces no risk of invalidating the k-anonymity property of the local data.*

Proof sketch. We need to show that the Synchronization step shown in Figure 3 does not break the k-anonymity properties of the local anonymization. First, we outline the basic setup and then go into detail about the k-anonymity property and discuss the t-closeness property thereafter.

The Synchronizer receives a k-anonymous and t-close data set from the Anonymizer component. The anonymization is valid w. r. t. to the local data only. Now, exchanging information about the ECs in an encrypted, thus privacy preserving way does not invalidate the k-anonymity property of the data. As outlined in subsection 3.2, the Synchronizer component demands the ECs to be identical in terms of the contained IDs. This is a special case of the multiset-equality property introduced in subsection 2.3. In case the synchronization step fails, the data is being handed back to the Anonymizer. The Anonymizer produces a more general version of the data, which will lead to larger ECs containing the same set of tuples as in the previous iteration plus additional tuples from another EC.

Theorem 2. *The Synchronizer component introduces no risk of invalidating the t-closeness property of the local data.*

Proof sketch. Following the idea of theorem 1, the Synchronizer receives a more general version of the local data in every iteration of the protocol. Due to the generalization property, outlined by Li et al. in [LLV07], the t-closeness is invariant to further generalizations. By using commutative encryption and exchanging information only on the IDs, neither site is able to learn anything about the frequency distribution of the SA-values. Following the same argument as Jiang et al. in [JC05], the data exchange step does not reveal information about the t-closeness of the local data.

Theorem 3. *The distributed anonymization protocol D2Pt computes a k-anonymous and t-close view of vertically partitioned data in an HBC model.*

Proof sketch. To show the compliance of the proposed protocol with the k-anonymity and t-closeness property, we have to analyze the result sets. First, we will elaborate on the k-anonymity property of the global data, e. g., the data depicted in Table 5c. As explained in subsection 3.6, each tuple contains a reference to the SA-tables, e. g., the GID. Since the GID is derived using the ECs constructed by the Datafly-algorithm, at least k tuples are referencing the same set of at least k SA-values. Following the works of Xiao et al. such fragmented data fulfills the k-anonymity property [XT06].

Using the fragmentation, we also maintain the t-closeness property on the local data throughout the release of the global data set. We will elaborate on this proposition by outlying a sketch of the steps necessary to prove this statement.

We start by reviewing Table 5c. As a result of the global anonymization process, we obtain three tables, e. g., a QID-table and two SA-tables. We will refer to the table containing the SA DISEASE as SA_0 and the table containing the SA TREATMENT as SA_1. We need to show the validity of the t-closeness for every SA in the set resulting from our protocol.

The two tables SA_0 and SA_1 contain a set of identical ECs. That is, the GID attribute references an identical set of individuals on both sites. Using the GID is the only way to reconstruct the original data. Therefore, the tables SA_0 and SA_1 can only be joined on their GID. Joining the tables on the GID results in a table $SA_\times$, by taking the Cartesian product of every EC from SA_0 with the matching EC of SA_1.

Given $SA_\times$, it is not useful to derive the t-closeness for SA_0 or SA_1. As the set includes a large number of tuples not included in the original data, called *phantom tuples* hereafter, the calculation of the t-closeness provides no information gain w. r. t. the original data. The amount of phantom tuples increases in the cardinality of the ECs of $SA_\times$. To estimate the amount of phantom tuples, we assume a minimal value for k-anonymity of 2. Thus, every EC introduces at least one phantom tuple for each existing tuple in the tables SA_0 and SA_1. In order to utilize the released data $SA_\times$ to infer information on the t-closeness, $SA_\times$ has to be modified. Specifically, the phantom tuples have to be removed, since they introduce perturbation of the data. This can be achieved based on the frequency distribution of SA_0 and SA_1. The goal is to reconstruct a set of tuples, termed $SA'_\times$, that includes one tuple for each individual in the original data.

We show the reduction of $SA_\times$ by revisiting our running example from Table 5c, using only the first EC with GID 1. We observe, that attribute value "Gastritis" of attribute DISEASE, would appear twice in $SA_\times$, e. g., "(Gastritis, Acid-reducing drug)" and "(Gastritis, Antacid)". One of these tuples must be a phantom tuple, since Table 5c contains "Gastritis" only once. Without background knowledge, we can only select one combination by chance. Next, we extend this example to the use of background knowledge.

Given the knowledge that Acid-reducing drugs are a promising treatment for Gastritis, we would select the corresponding tuple "(Gastritis, Acid-reducing drugs)" as candidate for the reconstruction. Therefore, we remove the identified phantom tuple "(Gastritis, Antacid)" from $SA_\times$. Applying this procedure to every EC results in $SA'_\times$, which contains every sensitive attribute value that appears in SA_0 or SA_1. It has to be noted, that the resulting tuples will not necessarily represent an individual present in the original table. Most important, the number of tuples in $SA'_\times$ equals the number of tuples in SA_0 or SA_1. Additionally, the frequency of every value in SA_0 or SA_1 is exactly the same as in $SA'_\times$. Thus, the frequency distribution of the attribute values remains unchanged. Since t-closeness is computed from the frequency distribution of attribute values, its result is the same compared to the t-closeness gained by the Synchronizer.

To sum up our proof sketch, we have shown that: (*1*) k-anonymity is maintained during the execution of the protocol in theorem 1, (*2*) t-closeness is maintained during the execution of the protocol in theorem 2 and (*3*) the result does adhere to the k-anonymity and t-closeness property, as shown in theorem 3.

As additional findings, we want to elaborate on properties regarding the calculation of the t-closeness for table $SA_\times$. First, joining the table SA_0 and SA_1 on the GID resulting in the table $SA_\times$. If the table $SA_\times$ contains ECs different in size, the resulting t-closeness computed over the frequency distribution of every attribute, will be different from the one computed on the local data sets. The bigger the difference in size between the smallest and largest EC, the more the t-closeness differs. Second, if the data in the table $SA_\times$ is skewed or correlated it is possible to infer information about the original distribution from the frequency distribution in the table $SA_\times$. For example, one could use the conditional probability measure to find the most likely join candidate among the tuple combinations in every EC. This allows the removal of impossible combinations in the whole table, e. g., the attribute value appears only once. Furthermore, our experiments indicate that it will likely be possible to reconstruct the original table, thus linking the SA-values. In subsection 3.6 this was identified as being a privacy threat. If the SA-values are unique a re-identification has appeared. However, the t-closeness is a privacy metric that does not address this kind of attack and thus cannot protect from it. The reconstructed table still adheres to t-closeness.

4 Experimental Evaluation

In this section, we evaluate our protocol D2P*t* presented in section 3. At first, we describe our experimental setup in subsection 4.1. Then, we compare the results with respect to the Synchronizer in section 4.2, i. e., the amount of synchronization attempts. We have also conducted an in-depth experimental analysis of the amount of data transmitted. We will leave out a detailed presentation of the results due to the focus of the paper being on the concepts of data anonymization and their adaption to a distributed environment. Finally, we analyze data utility of the published data in section 4.2.

4.1 Experimental setup

Prototypical implementation We implement D2P*t* as a prototype in Java. The prototype consists of a main program and two logical units representing the different sites. Each site holds an instance of an Anonymizer responsible for ensuring the k-anonymity and t-closeness properties. The Datafly-implementation is based on the Anonymization Toolbox of UT Dallas[2]. In contrast to the Anonymization Toolkit, which uses SQLite as a data store, we chose PostgreSQL to allow concurrent access to the database.

In a preparation step, we introduced a cache structure, thus, allowing us to reuse anonymization results from the Datafly-algorithm, therefore, significantly accelerating the process of data anonymization. In this paper, we exclude the preparation step from our measurements, since times needed for data anonymization and exchange are not in the focus of our evaluation. As mentioned in section 3 the anonymization algorithm is exchangeable, thus providing no use for a comparison of the anonymization algorithm. Rather, we focus on the simplicity and comprehensibility of our protocol. Since we are only interested in the utility of the anonymization results and the amount of data transmitted during the synchronization phase, we also exclude an evaluation of the commutative encryption scheme.

Data description We evaluate our protocol using the Adult Data Set[3], which is the de facto standard for benchmarking anonymization techniques [FTH$^+$11]. Out of the approx-

[2] http://cs.utdallas.edu/dspl/cgi-bin/toolbox/index.php

[3] https://archive.ics.uci.edu/ml/datasets/Adult

imately 50 000 entries forming the training data, we use only those containing no NULL values, i. e., 30 162 entries. As outlined in section 2.1 and section 2.1, we need a set of VGHs to model possible generalizations, as well as semantic relationship between attribute values. We adapt the VGHs presented by Fung et al. in [FWY05]. We classify the attributes contained in the Adult Data Set into QIDs and SAs. The QIDs are formed using the attributes *Workclass, Education, Marital Status, Relationship, Race, Sex* and *Native Country*. Likewise, the SAs are formed using the attributes *Occupation* and *Salary*. This is equivalent to the setting used by Li et al. [LLV07].

Workload For our test series we define the following schedule: We randomly assign each site one SA. Afterward, the QIDs attributes are distributed among the sites in every possible combination. Every combination of QIDs is evaluated with every combination of k, t values and every Synchronizer. For values of k we adapt the ones used by Jiang et al. in [JC05], namely, $k \in \{2, 5, 10, 20, 100\}$. Likewise, we use the values given for t by Li et al. [LLV07], i. e., $t \in \{0.1, 0.2\}$. To get a better understanding of the influence of high t values we add the value 0.7 and 1.0. The latter, allows us to simulate DPP$_2$GA using a specialized Synchronizer, without the need to modify out protocol. Thus we obtain $t \in \{0.15, 0.2, 0.7, 1.0\}$

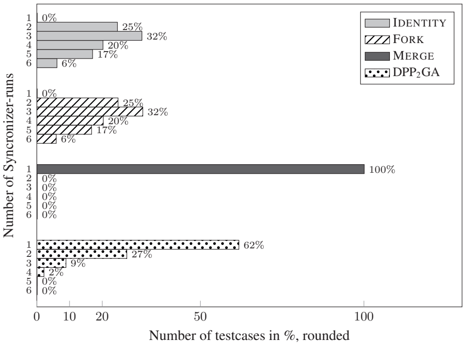
4.2 Experimental results

In the previous section we described the experimental setup. We will now analyze the impact of the k and t values on the resulting anonymization. Therefore, we differentiate our results into results called SAME and DIFF. Results stemming from SAME are obtained from initially identical anonymizations produced by the Anonymizer, i. e., the same ECs exist at both sites after the first anonymization phase. Results obtained from initially different anonymizations, will be called DIFF. The reason for the distinction between the two is given in Figure 4a. We observe that, given a t value of 0.15, the amount of initially identical anonymizations is almost 30 % for every k. That can be explained by the reduced amount of possible generalizations available for the strict privacy constraints of a small t. An initially identical anonymization renders the synchronization phase useless, since it evaluates to true after the first iteration. Since we want to observe the impact of t on the anonymization process at a whole, we distinguish between the cases SAME and DIFF. The following numbers apply to numbers obtained by DIFF unless otherwise stated. The term ALL will refer to the combined results of SAME and DIFF.

Synchronization attempts As outlined in subsection 3.5, situations occur when every Synchronizer except MERGE has to perform more than one generalization step. Since this is only the case for ECs stemming from DIFF, we directly observe the need for such a distinction. As expected, the MERGE-Synchronizer always performs its synchronization within one step. DPP$_2$GA also performs well during the synchronization phase, i. e., it needs more than three attempts in less than 2.1 % of all testcases. In contrast, the Syn-

	k					
t	2	5	10	20	50	100
0.15	29.4 %	29.4 %	29.4 %	30.1 %	30.1 %	30.1 %
0.2	5.5 %	5.5 %	5.5 %	6.0 %	6.0 %	6.0 %
0.7	1.2 %	2.2 %	2.5 %	2.7 %	3.2 %	3.5 %
1	1.6 %	2.2 %	2.5 %	2.7 %	3.2 %	3.5 %

(a) Percentage of same-share w. r. t. k and t



(b) Distribution of Syncronizer-runs

		t			
Testcase	Synchronizer	0.15	0.2	0.7	1.0
diff	Identity	1.00	1.00	0.97	0.97
	Fork	1.00	0.99	0.96	0.95
	Merge	0.99	0.84	0.63	0.63
	DPP$_2$GA	-	-	-	0.22
same	Identity	0.60	0.58	0.32	0.32
	Fork	0.60	0.58	0.32	0.32
	Merge	0.60	0.58	0.32	0.32
	DPP$_2$GA	-	-	-	0.32
all	Identity	0.88	0.97	0.95	0.95
	Fork	0.88	0.97	0.94	0.94
	Merge	0.87	0.82	0.63	0.62
	DPP$_2$GA	-	-	-	0.22

(c) Normalized discernibility metric dependent on the Synchronizer and t

Figure 4: Evaluation of D2Pt

chronizers Identity and Fork need more than three synchronization attempts in over 40 % of the cases. The only situations where Identity and Fork need less than three Synchronizer runs, is given by small t values, e. g., $\{0.15, 0.2\}$. That can be explained by the reduced amount of possible generalizations available for the strict privacy constraint of a small t.

Data utility As a measure to rate the utility of an anonymized data set we use the *discernibility metric (DM)*. This metric was introduced by Bayardo et al. [BA05]. It penalizes large ECs. Its result is the sum of these penalties. In order to gain comparable results, the sum has to be normalized. The evaluation on the utility dependent on the Synchronizer and t value is shown in Figure 4c.

As a pre-note, we want to emphasize, that results for DPP$_2$GA only apply for the case of $t = 1$, since the protocol does not support t-closeness. The results obtained by the Merge-Synchronizer are generally preferable to those of Identity and Fork. Surprisingly, Fork does not perform much better than Identity. While it is able to mark ECs in 61 % of its testcases, this only slightly affects the result of the DM. Not surprisingly, all Synchronizers show their worst performance for small t values. As noted before, a small t leads to maximal generalization with high probability.

Again, we can observe the use of splitting the result into the groups of same and diff. Clearly, every Synchronizer in the same group obtains the same utility, since no synchronization has to be performed after the first anonymization phase.

In summary, our results indicate that the utility of the data generated by D2P*t* is lower than that of DPP$_2$GA. However, this also results from the use of t-closeness as a privacy metric and its higher demands on the structure of ECs.

5 Related work

The protocol DPP$_2$GA has been criticized for relaxing the notion of SMC. Therefore, Jiang et al. [JC06] extended DPP$_2$GA to the protocol DkA to comply with the rules of SMC. Both protocols perform a *bottom-up generalization* by gradually generalizing QID values until achieving k-anonymity.

Mohammed et al. [MFD11] propose a *top-down* variant of a distributed k-anonymization protocol. This protocol, iteratively specializes maximally generalized QID values, thus producing an approximate optimal k-anonymity.

Recent work by Kohlmayer et al. [KPEK13] presents a study on state-of-the-art anonymization algorithms applied to a distributed environment [KPEK13]. The proposed framework is capable of dealing with horizontally and vertically partitioned data sets. One of the concepts discussed by Kohlmayer et al. is the concept of t-closeness. However, as Kohlmayer et al. pointed out, their framework is not capable of dealing with numerical values in case of t-closeness. Furthermore, it was not shown if the framework respects the threats imposed by the background-join attack introduced by Fang et al. [FAN11].

Closely related to the adaption of privacy metrics to a distributed environment is the question of dealing with multiple sensitive attributes (MSA). This setting is imposed by a distributed scenario where each participating institution holds at least one SA.

The problem of MSA does not affect k-anonymization algorithms, since k-anonymity does not require the SAs to satisfy any properties. The concept of ℓ-diversity demands SAs values to be "well-represented" [MKGV07]. Machanavajjhala et al. identify the problem of MSA, but omit a detailed discussion. Recent work by Das et al. [DB12] addresses the problem of MSA and provide a solution specifically tailored for ℓ-diversity. The solution of Das et al., *Decomposition+*, is related to ours, since it also fragments the data.

Data fragmentation has been discussed before in the context of k-anonymization. It is achieved by the *Anatomy*-algorithm developed by Xiao et al. [XT06]. Recently, Fang et al. [FAN11] addressed the problem of MSA for centralized t-closeness. Providing a fragmenting solution, that analyzes the information loss imposed by the fragmentation.

6 Conclusion & future work

Throughout this paper, we have shown a simple yet effective method to enhance the distributed k-anonymization algorithm DPP$_2$GA with the privacy requirements of t-closeness. Using the MERGE-Synchronizer, we were able to significantly enhance the utility of our distributed anonymization protocol.

Moreover, the results we have gathered so far, indicate that the globally t-close table is identical to the one produced by a centralized instance of the Datafly-algorithm. However, a more detailed discussion on this assumption is subject to future research.

Nevertheless, our evaluation shows that DPP$_2$GA achieved better utility. Therefore, it should be noted, that the concept of t-closeness itself lowers utility of the data. In fact, Li et al. criticized that it “substantially limits the amount of useful information that can be extracted from the released data” [LLV10].

The work presented in this paper focuses on the simplicity of the protocol, thus relaxing some security requirements typically imposed by the notion of SMC. Future work could focus on allowing distributed t-closeness under the strict requirements of SMC. Furthermore, our current protocol is limited to a two-party design, thus allowing only two sites to jointly compute an anonymization. It would be a challenging task to adopt our protocol to an n-party environment, i. e., an arbitrary number of participating sites. Finally, it would be of great interest to investigate the benefits of bottom-up and top-down approaches and their application in an distributed environment.

References

[AFK$^+$05] G. Aggarwal, T. Feder, K. Kenthapadi, R. Motwani, R. Panigrahy, D. Thomas, and A. Zhu. Approximation Algorithms for k-Anonymity. *Journal of Privacy Technology*, 2005.

[BA05] R. J. Bayardo and R. Agrawal. Data Privacy through Optimal k-Anonymization. In *Proceedings of the 21st International Conference on Data Engineering*, 2005.

[CDCdVFS07] V. Ciriani, S. De Capitani di Vimercati, S. Foresti, and P. Samarati. Microdata protection. In *Secure Data Management in Decentralized Systems*, 2007.

[CT13] C. W. Clifton and T. Tassa. On syntactic anonymity and differential privacy. In *ICDE Workshop on Privacy-Preserving Data Publication and Analysis*, 2013.

[Dal77] T. Dalenius. Towards a methodology for statistical disclosure control. *Statistisk Tidskrift*, 15(15):429–444, 1977.

[DB12] D. Das and D. K. Bhattacharyya. Decomposition+: Improving ℓ-Diversity for Multiple Sensitive Attributes. In *Advances in Computer Science and Information Technology. Computer Science and Engineering*, pages 403–412. 2012.

[FAN11] Y. Fang, M. Ashrafi, and S. Ng. Privacy beyond Single Sensitive Attribute. In *Database and Expert Systems Applications*. 2011.

[FED94] Federal Committee on Statistical Methodology. *Report on Statistical Disclosure Limitation Methodology*, 1994.

[FTH$^+$11] B. C. M. Fung, T. Trojer, P. C. K. Hung, L. Xiong, Khalil A.-H., and R. Dssouli. Service-Oriented Architecture for High-Dimensional Private Data Mashup. *IEEE Transactions on Services Computing*, 2011.

[FWY05] B. C. M. Fung, K. Wang, and P. S. Yu. Top-down specialization for information and privacy preservation. In *Proceedings of the 22nd International Conference on Data Engineering*, pages 205–216, 2005.

[Gol04] Oded Goldreich. *Foundations of Cryptography: Basic Applications*, volume 2. Cambridge University Press, 1 edition, 07 2004.

[JC05] W. Jiang and C. W. Clifton. Privacy-Preserving Distributed k-Anonymity. In *Working Conference on Data and Applications Security*, 2005.

[JC06] Wei Jiang and Christopher W. Clifton. A secure distributed framework for achieving k-anonymity. *The VLDB Journal*, 15:316–333, 11 2006.

[KPEK13] F. Kohlmayer, F. Prasser, C. Eckert, and K. A. Kuhn. A flexible approach to distributed data anonymization. *Journal of Biomedical Informatics*, 2013.

[LeF07] K. LeFevre. *Anonymity In Data Publishing And Distribution.* PhD thesis, 2007.

[LLV07] N. Li, T. Li, and S. Venkatasubramanian. t-Closeness: Privacy Beyond k-Anonymity and ℓ-Diversity. In *Proceedings of the 23rd International Conference on Data Engineering*, 2007.

[LLV10] Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian. Closeness: A New Privacy Measure for Data Publishing. *IEEE Transactions on Knowledge and Data Engineering*, 22:943–956, 2010.

[MFD11] Noman Mohammed, Benjamin C. M. Fung, and Mourad Debbabi. Anonymity meets game theory: secure data integration with malicious participants. *The VLDB Journal*, 20:567–588, 2011.

[MKGV07] Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramaniam. ℓ-diversity: Privacy beyond k-anonymity. *ACM Transactions on Knowledge Discovery from Data*, 1, 03 2007.

[MW04] Adam Meyerson and Ryan Williams. On the complexity of optimal K-anonymity. In Catriel Beeri and Alin Deutsch, editors, *PODS*, pages 223–228. ACM, 2004.

[RTG00] Y. Rubner, C. Tomasi, and L. J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 11 2000.

[Sam01] Pierangela Samarati. Protecting Respondents' Identities in Microdata Release. *IEEE Transactions on Knowledge and Data Engineering*, 13:1010–1027, 2001.

[SuLMA80] Adi Shamir and Ronald L. Rivest und Leonard M. Adleman. *Mental poker*, pages 37–43. Springer US, 1980.

[Swe97] Latanya Sweeney. Guaranteeing anonymity when sharing medical data, the Datafly System. *Proceedings : a conference of the American Medical Informatics Association*, pages 51–55, 10 1997.

[Swe02a] Latanya Sweeney. Achieving k-Anonymity Privacy Protection Using Generalization and Suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 10(5):571–588, 10 2002.

[Swe02b] Latanya Sweeney. k-Anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems*, 2002.

[US96] 104th Congress. *Health insurance portability and accountability act*, 1996.

[XT06] Xiaokui Xiao and Yufei Tao. Anatomy: Simple and Effective Privacy Preservation. In *Proceedings of the 32nd International Conference on Very Large Data Bases*, pages 139–150. ACM, 2006.