Journal of Privacy and Confidentiality (2016) **7**, Number 2, 127–158

Heterogeneous Differential Privacy

Mohammad Alaggan,*† Sébastien Gambs‡ and Anne-Marie Kermarrec§

Abstract. The massive collection of personal data by personalization systems has rendered the preservation of privacy of individuals more and more difficult. Most of the proposed approaches to preserve privacy in personalization systems usually address this issue uniformly across users, thus ignoring the fact that users have different privacy attitudes and expectations (even among their own personal data). In this paper, we propose to account for this non-uniformity of privacy expectations by introducing the concept of heterogeneous differential privacy. This notion captures both the variation of privacy expectations among users as well as across different pieces of information related to the same user. We also describe an explicit mechanism achieving heterogeneous differential privacy, which is a modification of the Laplacian mechanism by Dwork, McSherry, Nissim and Smith. In a nutshell, this mechanism achieves heterogeneous differential privacy by manipulating the sensitivity of the function using a linear transformation on the input domain. Finally, we evaluate on real datasets the impact of the proposed mechanism with respect to a semantic clustering task. The results of our experiments demonstrate that heterogeneous differential privacy can account for different privacy attitudes while sustaining a good level of utility as measured by the recall for the semantic clustering task.

1 Introduction

The amount of personal information about individuals exposed on the Internet is increasing by the second. While such data may be used for recommendation and personalization purposes (Bertier et al., 2010; Zeng et al., 2012; Zhou et al., 2012; Wen and Lin, 2010; Liu et al., 2004), this also raises serious privacy concerns. At first, leveraging personal information to enhance the user experience through personalization services might seem contradictory with the preservation of the privacy of users of such systems. However in recent years, several approaches have been proposed to rely on Privacy-Enhancing Technologies (PETs), whose aim is to preserve privacy while maintaining a good level of utility for the proposed personalization service (Toch et al., 2012; Alaggan et al., 2011; 2012; McSherry and Mironov, 2009; Venkatasubramanian, 2008). One popular approach whose objective is to provide strong privacy guarantees despite auxiliary information that the adversary could have is the concept of differential privacy (Dwork, 2008; Mironov et al., 2009; McGregor et al., 2011; McSherry and Mironov, 2009; Dwork and Naor, 2010; Dwork et al., 2006; Beimel et al., 2011).

Most of these approaches implicitly assume homogeneity by considering that users

*Univ Lyon, Inria, INSA Lyon, CITI, F-69621 Villeurbanne, France
†Helwan University, Cairo, Egypt mailto:mohammad.alaggan@inria.fr
‡Université du Québec à Montréal (UQAM), Montréal, Canada mailto:gambs.sebastien@uqam.ca
§Inria, Rennes, France mailto:anne-marie.kermarrec@inria.fr

have uniform privacy requirements. However, in an environment composed of a myriad of communities, such as the Internet, it is highly plausible that users have heterogeneous privacy attitudes and expectations. For instance, consider a collaborative social platform in which each user is associated to a profile (*e.g.*, a set of URLs that a user has tagged in a system such as Delicious[1]). It is natural to expect that for a particular user some items in his profile are considered more sensitive by him than others, thus calling for a system that can deal with different privacy requirements across items. Similarly, Alice might be more conservative about her privacy than Bob, requiring different privacy requirements across users.

This non-uniformity of privacy attitudes has been acknowledged by major social networking sites (Preibusch and Beresford, 2009; Liu et al., 2011). For instance in Facebook, a user can now set individual privacy settings for each item in his profile. However in this particular example, privacy is mainly addressed by restricting, through an access-control mechanism, who is allowed to access and view a particular piece of information. Our approach can be considered to be orthogonal but complementary to access-control. More precisely, we consider a personalized service, such as a recommendation algorithm, and we enforce the privacy requirements of the user on its output. Heterogeneous privacy requirements might also arise with respect to pictures, depending on the location in which the picture was taken or the persons appearing on it (Liu et al., 2011). In the future, users are likely to expect item-grained privacy for other services[2].

Furthermore, as highlighted by Zwick and Dholakia (1999) and as evidenced by anthropological research, privacy attitudes are highly dependent on social and cultural norms. A similar point was raised in 2007 by Zhang and Zhao in a paper on privacy-preserving data mining (Zhang and Zhao, 2007) in which they mentioned that in practice it is unrealistic to assume homogeneous privacy requirements across a whole population. In particular, their thesis is that enforcing the same privacy level across all users and for all types of personal data could lead to an unnecessary degradation of the performance of such systems as measured in terms of accuracy. More specifically, enforcing the same privacy requirements upon all users (even those who do not require it) might degrade the performance in comparison to a system in which strict privacy requirements are only taken into account for those who ask for it. The same type of argument can also be made for different items of the same user. Hence, designing a system supporting heterogeneous privacy requirements could lead to a global improvement of the performance of this system as compared to a homogeneous version. Therefore, the main challenge is to be able to account for the variety of privacy requirements when leveraging personal data for recommendation and personalization.

In this paper, we address this challenge through the introduction of the concept

[1] http://del.icio.us/

[2] Note that systems supporting item-grained privacy can also provide user-grained privacy (*i.e.*, for instance by setting the privacy level of all items in some user's profile to the same value in the privacy setting of this user), and therefore the former can be considered as a generalization of the latter. However, this assumes that the privacy weights have a global meaning across the entire system, and are not defined only relative to a user.

of *heterogeneous differential privacy*, which considers that the privacy requirements are not homogeneous across users and items from the same user (thus providing item-grained privacy). This notion can be seen as an extension of the concept of differential privacy introduced originally by Dwork et al. (2006) in the context of databases. We also describe an explicit mechanism achieving heterogeneous differential privacy, which we coin as the "stretching mechanism". We derive a bound on the distortion introduced by our mechanism, which corresponds to a distance between the expected output of the mechanism and the original value of the function to be computed. Finally, we conduct an experimental evaluation of our mechanism on a semantic clustering task using real datasets. The results obtained show that the proposed approach can still sustain a high utility level (as measured in terms of recall) while guaranteeing heterogeneous differential privacy.

The outline of the paper is as follows. First, in Section 2, we describe the background of differential privacy as well as some preliminaries on matrices and sets necessary to understand our work. Afterwards in Section 3, we introduce the novel concept of heterogeneous differential privacy along with the description of an explicit mechanism achieving it. Then, we assess experimentally the impact of the proposed mechanism by evaluating it on a semantic clustering task in Section 4. In Section 5, we present the related work on heterogeneous privacy mechanisms before concluding with a discussion on the actual limitations of the approach as well as possible extensions in Section 6.

2 Background

In this section, we briefly introduce the background on differential privacy (Section 2.1) as well as some basic notions that are necessary to understand the concept of heterogeneous differential privacy (Section 2.2).

2.1 Differential Privacy

We begin with providing some background of differential privacy, which was originally introduced by Dwork et al. (2006) in the context of statistical databases. The main guarantee provided by this approach is that if a differentially private mechanism is applied on a database composed of the personal data of individuals, no output would become significantly more (or less) probable whether or not a participant removes this particular data from the dataset. In a nutshell, it means that for an adversary observing the output of the mechanism, the advantage gained from the presence (or absence) of a particular individual in the database is negligible. This statement is a statistical property about the behavior of the mechanism (*i.e.*, function) and holds independently of the auxiliary knowledge that the adversary might have gathered. More specifically, even if the adversary knows the whole database but one individual row, a mechanism satisfying differential privacy still protects the privacy of this row. The parameter ε is public and may take different values depending on the application (for instance it could be 0.01, 0.1, 0.25 or even 2). While it is sometimes difficult to grasp the intuition about

the significance of a particular value for ε (Lee and Clifton, 2011), a smaller value of ε implies a higher privacy level.

Differential privacy was originally designed for ensuring privacy to individuals who have contributed with their personal data to the construction of a statistical database. In this setting, each individual is a row (*i.e.*, coordinate) in this database (*i.e.*, vector). Differential privacy guarantees that *almost* no difference will be observed to the output of the query performed on the database, whether or not the individual (a single row) has contributed to the database by submitting his data, and therefore this information is considered as being protected.

When the database is the profile of a user, which is a vector of items (sometimes called *the micro-data setting*), the whole vector (*i.e.*, database) is owned by a single individual. This difference impacts the interpretation that can be done when speaking about *protecting the privacy of this individual*. In particular, contrary to the first setting of statistical database, an individual does not have the choice to submit or not his data. Rather, if he chooses not to use his profile as input to the collaborative social system, then he will not benefit from the service. However in this new setting, the user is still left with the possibility of selecting a subset of items in his profile before participating. In this case, the main objective of differential privacy is to ensure that when a user adds or removes a single item from his profile, this has a small effect on the output of the computation. However, one caveat is that if the profile of the user contains nothing but items related to a particular sensitive topic (*e.g.*, cancer), then in order to get at least a little bit of utility that information has to be leaked. This observation is in line with the impossibility result of Dwork and Naor stating that if a privacy-preserving mechanism provides any utility, then it has to cause a privacy breach whose magnitude is at least proportional to the min-entropy of the utility (Dwork and Naor, 2010). Thus, this limitation is true for any possible privacy-preserving mechanism and is not inherent to the micro-data setting (*i.e.*, this limitation also holds for the database setting).

The difference of a single row between two profiles can be defined formally through the concept of *neighboring profiles*. Each user is associated with a profile representing his personal data, which can be defined as a vector in $\mathbb{R}^n$ (for some n fixed for all users across the system). This representation is generic enough to encompass a variety of possible user profiles. For instance, restricting the domain to $\{0,1\}^n$ can be used to represent a binary string (which is a universal representation) or a subset of items of a global domain of items.

Definition 1 (Neighboring profile)**.** *Two profiles $\vec{d}, \vec{d}^{(i)} \in \mathbb{R}^n$ are said to be neighbors if there exists an item $i \in \{1, \ldots, n\}$ such that $d_k = d_k^{(i)}$ for all items $k \neq i$. This neighboring relation is denoted by $\vec{d} \sim \vec{d}^{(i)}$.*

An equivalent definition states that $\vec{d}$ and $\vec{d}^{(i)}$ are neighbors if they are identical except for the i-th coordinate. For instance, the profiles $(0,1,2)$ and $(0,2,2)$ are neighbors while the profiles $(0,1,2)$ and $(0,2,3)$ are not. Differential privacy can be defined formally in the following manner.

Definition 2 (ϵ-differential privacy (Dwork et al., 2006))**.** *A randomized function $\mathcal{M}$:*

$\mathbb{R}^n \to \mathbb{R}$ *is said to be* ε*-differentially private if for all neighboring profiles* $\vec{d} \sim \vec{d}^{(i)} \in \mathbb{R}^n$*, and for all outputs* $t \in \mathbb{R}$*, the following statement holds:*

$$\Pr[\mathcal{M}(\vec{d}) = t] \leqslant \exp(\varepsilon) \Pr[\mathcal{M}(\vec{d}^{(i)}) = t] \ , \tag{1}$$

in which exp *refers to the exponential function.*

Differential privacy aims at reducing the contribution that any single coordinate of the profile can have on the output of a function. The maximal magnitude of such contribution is captured by the notion of (global) *sensitivity*.

Definition 3 (Global sensitivity (Dwork et al., 2006))**.** *The global sensitivity* $S(f)$ *of a function* f *is the maximum absolute difference obtained on the output over all neighboring profiles:*

$$S(f) = \max_{\vec{d} \sim \vec{d}^{(i)}} |f(\vec{d}) - f(\vec{d}^{(i)})| \ , \tag{2}$$

in which $\vec{d} \sim \vec{d}^{(i)}$ *means that* $\vec{d}$ *and* $\vec{d}^{(i)}$ *are neighboring profiles (*cf. *Definition 1).*

Dwork et al. (2006) proposed a technique called the *Laplacian mechanism* that achieves ε-differential privacy by adding noise to the output of a function proportional to its global sensitivity. The noise is distributed according to the Laplace distribution (with PDF $\frac{1}{2\sigma} \exp(-|x|/\sigma)$, in which $\sigma = S(f)/\varepsilon$ is a scale parameter).

The novel mechanism that we propose in this paper (to be detailed later) achieves heterogeneous differential privacy by modifying the sensitivity of the function to be released (and therefore the function itself) before applying the standard Laplacian mechanism.

2.2 Preliminaries

Before delving into the details of our approach, we need to briefly introduce some preliminary notions on matrices and sets such as the concept of *shrinkage matrix* (Jeffrey, 2010). A shrinkage matrix is a linear transformation that maps a vector to another vector with less magnitude, possibly distorting it by changing its direction.

Definition 4 (Shrinkage matrix)**.** *A matrix* A *is called a shrinkage matrix if and only if* $A = \mathrm{diag}(\alpha_1, \ldots, \alpha_n)$ *such that each diagonal coefficient is in the range* $0 \leqslant \alpha_i \leqslant 1$*.*

For example, the matrix

$$\begin{pmatrix} 0.7 & 0 & 0 \\ 0 & 0.3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

is a shrinkage matrix.

Definition 5 (Semi-balanced set)**.** *A set* $D \subseteq \mathbb{R}^n$ *of column vectors is semi-balanced if and only if for all shrinkage matrices* $A = \mathrm{diag}(\alpha_1, \ldots, \alpha_n)$*, and for all* $\vec{x} \in D$*, we have* $A\vec{x} \in D$*.*

For instance, the set

$$\{\vec{x} = (x_1, x_2) \in \mathbb{R}^2 \mid 0 < x_1, x_2 < 1\}$$

is a semi-balanced set that can be visualized as a square from $(0, 0)$ to $(1, 1)$ in the Euclidean plane.

We will also use the following result later in our work.

Proposition 1 (Semi-balanced sets are closed under shrinkage)**.** *If D is a semi-balanced set and A is a shrinkage matrix, then AD is also a semi-balanced set.*

Proof 1. *AD is semi-balanced if and only if for any shrinkage matrix B the following is true: $B(AD) \subseteq AD$. Indeed, since B is a shrinkage matrix and D is a semi-balanced set, then $BD \subseteq D$. By multiplying both sides by a shrinkage matrix A, we obtain $ABD \subseteq AD$, which by the fact that A and B are commutative implies that $BAD \subseteq AD$.*

□

3 Heterogeneous Differential Privacy

In this section, we introduce the novel concept of *heterogeneous differential privacy* (HDP). We start by giving the necessary definitions in Section 3.1, before describing in Section 3.2 how to construct the Stretching Mechanism, which ensures heterogeneous differential privacy. More precisely, we first detail how to construct the privacy-preserving estimator in Section 3.2.1. Afterwards, we discuss why and how the privacy vector expressing the privacy expectations of a user should also be kept private in Section 3.2.3. Finally, an upper bound on the distortion induced by the Stretching Mechanism is provided in Section 3.2.4.

3.1 Definitions

We now define HDP-specific notions such as the concept of *privacy vector*, which is a key notion in HDP. This vector contains the privacy requirements of each coordinate (*i.e.*, item) in the input profile (*i.e.*, vector) of a user, and is defined as follows.

Definition 6 (Privacy vector)**.** *Given a user and his profile $\vec{d} \in D$ in which D is a semi-balanced set of column vectors composed of n coordinates, let $\vec{v} \in [0, 1]^n$ be the privacy vector associated with the userand his profile $\vec{d}$. The owner of item d_i is responsible for choosing the privacy weight v_i associated to this item (In Remark 2, we discuss what should be the default weight if it is not explicitly provided by the user). A privacy weight v_i of zero corresponds to* absolute privacy *while a value of* 1 *refers to* standard privacy*, which in our setting directly correspond to the classical definition of ε-differential privacy.*

The mere existence of the privacy vector introduces potential privacy breaches, which means that this vector should also be protected. Thus, we need to ensure that in addition

to the profile, the privacy vector $\vec{v}$ also remains private, such that each entry v_i of this vector should only be known by its owner. Otherwise, the knowledge of a privacy weight of a particular item might leak information about the profile itself. For instance, learning that some items have a high privacy weight may reveal that the user has high privacy expectations for and is therefore interested in this specific type of data. We define *heterogeneous differential privacy* in the following manner.

Definition 7 ((Heterogeneous) $(\varepsilon, \vec{v})$-differential privacy)**.** *For all semi-balanced sets D, a randomized function $\mathcal{M} : D \to \mathbb{R}$ is said to be $(\varepsilon, \vec{v})$-differentially private if for all items i, for all neighboring profiles $\vec{d} \sim \vec{d}^{(i)} \in D$, and for all outputs $t \in \mathbb{R}$, the following statement holds:*

$$\Pr[\mathcal{M}(\vec{d}) = t] \leqslant \exp(\varepsilon v_i) \Pr[\mathcal{M}(\vec{d}^{(i)}) = t], \tag{3}$$

in which exp *refers to the exponential function.*

Since a privacy weight $v_i \leqslant 1$, heterogeneous differential privacy implies the standard notion of ε-differential privacy as shown by the following remark.

Remark 1 (Equivalence of $(\varepsilon, \vec{v})$-DP and ε-DP.)**.** *Let $\overline{\varepsilon} = \varepsilon\overline{v}$ and $\underline{\varepsilon} = \varepsilon\underline{v}$, such that $\overline{v} = \max_i v_i$ (the maximum privacy weight) and $\underline{v} = \min_i v_i$ (the minimum privacy weight). Then, we have: $\underline{\varepsilon}$-DP $\implies$ $(\varepsilon, \vec{v})$-DP and $(\varepsilon, \vec{v})$-DP $\implies$ $\overline{\varepsilon}$-DP. As a consequence, $(\varepsilon, \vec{1})$-DP holds* if and only if *ε-DP also holds, in which $\vec{1} = (1, \cdots, 1)$.*

Finally, we rely on a variant of the notion of global sensitivity, implicitly introduced in Dandekar et al. (2011, Lemma 1), that we call *modular global sensitivity*.

Definition 8 (Modular global sensitivity Dandekar et al. (2011))**.** *The modular global sensitivity $S_i(f)$ is the global sensitivity of f when $\vec{d}$ and $\vec{d}^{(i)}$ are neighboring profiles that differ on exactly the item i.*

In a nutshell, the modular global sensitivity reflects the maximum difference that a *particular item i* can cause by varying its value (over its entire domain) while keeping all other items fixed.

3.2 The Stretching Mechanism

Thereafter, we describe a generic mechanism achieving heterogeneous differential privacy that we coin as the *Stretching Mechanism*. We assume that the privacy preferences for each item are captured through a privacy vector $\vec{v}$ (*cf.* Definition 6). Given an arbitrary *total* function $f : D \to \mathbb{R}$, in which D is a semi-balanced set of columns vectors of n coordinates, and whose global sensitivity $S(f)$ is finite, we construct a randomized function $\mathcal{SM}(\vec{d}, \vec{v}, \varepsilon)$ *estimating* f while satisfying $(\varepsilon, \vec{v})$-differential privacy. The Stretching Mechanism is described below in Algorithm 1.

Before delving into the details of this method, we provide a little intuition on how and why it works. A lemma in Dandekar et al. (2011, Lemma 1) asserts that the

Algorithm 1 The StretchingMechanism

procedure StretchingMechanism(f, $\vec{d}$, $\vec{v}$, ε, δ)
  for each `i in` $\{1, \ldots, \dim(\vec{v})\}$ **do** ▷ Compute a vector $\vec{w}$ from the privacy vector $\vec{v}$
    Let $R_\alpha(\vec{x}) : D \to \mathbb{R}$ be such that $R_\alpha(\vec{x}) = f(\vec{x}_1, \ldots, \alpha\vec{x}_i, \ldots, \vec{x}_{\dim(D)})$, in which $\alpha \in \mathbb{R}$
    Let $\alpha \leftarrow 1$
    while $S_i(R_\alpha) > v_i S(f)$ **and** $\alpha \geq 0$ **do**
      Set $\alpha \leftarrow \alpha - \delta$
    end while
    if $\alpha < 0$ **then**
      Set $\vec{w}_i \leftarrow 0$
    else
      Set $\vec{w}_i \leftarrow \alpha$
    end if
  end for
  return $f(\mathrm{diag}(\vec{w}) \cdot \vec{d}) + \mathrm{LaplaceRandomness}(0, S(f)/\varepsilon)$ ▷ Laplacian Mechanism
end procedure

Laplacian mechanism $\mathcal{M}(\vec{d}) = f(\vec{d}) + \mathsf{Lap}(\sigma)$ with mean 0 and standard deviation σ provides $\Pr[\mathcal{M}(\vec{d}) = t] \leqslant \exp(\varepsilon_i) \Pr[\mathcal{M}(\vec{d'}^{(i)}) = t]$, in which $\varepsilon_i = S_i(f)/\sigma$. In other words, differential privacy can be achieved by setting the perturbation induced by the Laplacian mechanism to be proportional to the modular global sensitivity (Dandekar et al., 2011) instead of the standard global sensitivity. Therefore, a natural approach for enforcing heterogeneous differential privacy is to manipulate the modular global sensitivity $S_i(f)$ by modifying the function f itself.

3.2.1 Constructing the Estimator

Let $T : [0,1]^n \to \mathbb{R}^{n \times n}$ be a function taking as input a privacy vector $\vec{v}$ and returning as output a shrinkage matrix, with the property that $T(\vec{1}) = I$, such that I is the identity matrix and $\vec{1} = (1, \cdots, 1)$. Let also R be a mapping sending a function $f : D \to \mathbb{R}$ and a privacy vector $\vec{v} \in [0,1]^n$ to the function $R(f, \vec{v}) : D \to \mathbb{R}$ with $R(f, \vec{v})(\vec{d}) = f(T(\vec{v}) \cdot \vec{d})$. Recall that the Laplace distribution centered at 0 with scale parameter σ has the following probability density function

$$h(x) = \frac{1}{2\sigma} \exp(-|x|/\sigma) \ . \tag{4}$$

Finally, let N be a Laplacian random variable with parameter $\sigma = \sigma(f, \varepsilon) = S(f)/\varepsilon$, in which $S(f)$ refers to the global sensitivity of the function f and ε the privacy parameter. The following statement proves that this *Stretching Mechanism* R satisfies heterogeneous differential privacy.

Theorem 1 (Achieving HDP via the stretching mechanism)**.** *Given a privacy vector $\vec{v}$, if the function $T(\vec{v})$ satisfies $S_i(R(f,\vec{v})) \leqslant v_i S(f)$ then the randomized function $\mathcal{SM}(\vec{d},\vec{v},\varepsilon) = R(f,\vec{v})(\vec{d}) + N$ satisfies $(\varepsilon,\vec{v})$-differential privacy.*

Proof 2. *For all two neighboring profiles $\vec{d}, \vec{d}^{(i)}$, and for all outputs $t \in \mathbb{R}$ of the function f we have*

$$\begin{aligned}
\frac{\Pr[\mathcal{SM}(\vec{d},\vec{v},\varepsilon)=t]}{\Pr[\mathcal{SM}(\vec{d}^{(i)},\vec{v},\varepsilon)=t]} &= \frac{h(t-R(f,\vec{v})(\vec{d}))}{h(t-R(f,\vec{v})(\vec{d}^{(i)}))} \\
&\leqslant \exp\left(\frac{\varepsilon|R(f,\vec{v})(\vec{d}) - R(f,\vec{v})(\vec{d}^{(i)})|}{S(f)}\right) \\
&\leqslant \exp\left(\frac{\varepsilon S_i(R(f,\vec{v}))}{S(f)}\right) \\
&\leqslant \exp\left(\frac{\varepsilon v_i \cancel{S(f)}}{\cancel{S(f)}}\right) = \exp(\varepsilon v_i),
\end{aligned}$$

in which $h(\cdot)$ is defined in Equation (4), thus proving the result. □

In a nutshell, $T(\vec{v})$ is a shrinkage matrix, whose shrinking factor in each coordinate is computed independently of all other coordinates. More precisely, the shrinking factor for a particular item depends only on the privacy weight associated to this coordinate. The value used by the mechanism is the lowest amount of shrinkage (*i.e.*, distortion) still achieving the target modular global sensitivity of that coordinate. In the following section we provide an explicit construction of $T(\vec{v})$ for which we prove that by Lemma 1 the condition of Theorem 1 is satisfied, and therefore that $\hat{f}$ achieves $(\varepsilon,\vec{v})$-differential privacy.

3.2.2 Computing the Shrinkage Matrix

The HDP mechanism $\hat{f}(\vec{d},\vec{v},\varepsilon)$ adds Laplacian noise to a modified function $R(f,\vec{v})(\vec{d}) = f(T(\vec{v})\cdot\vec{d})$. In this section, we specify how to construct $T(\vec{v})$ such that $\hat{f}$ satisfies HDP. Thereafter, we use R to denote $R(f,\vec{v})$ for the sake of simplicity. Let $T(\vec{v}) = \mathrm{diag}(\vec{w})$ for some $\vec{w} \in [0,1]^n$ to be computed from the privacy vector $\vec{v}$ and $S(R,\vec{w})$ be the sensitivity of $R = f(T(\vec{v})\cdot\vec{d}) = f(\mathrm{diag}(\vec{w})\cdot\vec{d})$ given $\vec{w}$. Similarly, let $S_i(R,\vec{w})$ be the modular global sensitivity of R given $\vec{w}$. We denote by $(\vec{w}_{-i}, w_i')$ the vector resulting from replacing the item w_i in $\vec{w}$ to w_i' (*e.g.*, $(\vec{1}_{-i}, w_i) = (1,\ldots,w_i,\ldots,1)$). Each w_i can be computed from v_i by solving the following optimization problem:

$$\begin{array}{ll} \max & w_i \ , \\ \text{subject to:} & S_i(R,(\vec{1}_{-i},w_i)) \leqslant v_i S(f) \ . \end{array} \tag{5}$$

Note that a solution satisfying this constraint always exists and can be reached by setting w_i to 0. The w_i's are never released after they have been computed locally by the rightful owner, and the modular global sensitivity $S_i(R)$ is only used in the proof

and is not revealed to the participants, in the same manner as the noise generated. The participants only have the knowledge of the global sensitivity $S(f)$. Thus, the only way in which the profile $\vec{d}$ could leak is through its side effects to the output, which we prove to achieve ε-DP in Theorem 2.

Lemma 1. *If* $T(\vec{v}) = \text{diag}(\vec{w})$ *such that for all* i*:*

$$S_i(R, (\vec{1}_{-i}, w_i)) \leqslant v_i S(f) \tag{6}$$

(the constraint of (5)*) then* R *satisfies:*

$$S_i(R, \vec{w}) \leqslant v_i S(f) \tag{7}$$

for all i*.*

Proof 3. *See Appendix.* □

Algorithm 1 assumes that there is an efficient algorithm to compute the modular global sensitivity of $S_i(R, (\vec{1}_{-i}, w_i))$. This assumption is based on the intuition that if there is such an algorithm for f, then it should be easy to modify it (probably in a non-black-box manner) to accommodate for one scaled component. Given this assumption, the Algorithm 1 solves the optimization problem in a suboptimal manner using a parameter $0 < \delta < 1$ defining a tradeoff. If δ is too big, there is a possibility that the weight is far from optimal but still within δ distance of it. On the other hand, if δ is too small, the efficiency of this computation will be impacted. In particular if $\delta = 1/k$ for any positive integer k, then in the worst case $O(k)$ steps will be needed, each of which needs to compute the modular global sensitivity.

3.2.3 Hiding the Privacy Vector

By themselves, the privacy weights could lead to a privacy breach if they are released publicly (Ghosh and Roth, 2011; Dandekar et al., 2011). For instance, learning that the user has set a high weight on a particular item might indicate that the user possesses this item on his profile and that he has a high privacy expectation about it. Thus, the impact of the privacy weights on the observable output of the mechanism should be characterized. Moreover, when a user adds an item to his profile, it is likely that he will also simultaneously modify the corresponding privacy weight. That is, both the item and its privacy weight might change *simultaneously*.

The following theorem states that for all neighboring profiles $\vec{d} \sim \vec{d}^{(i)}$ and neighboring privacy vectors $\vec{v} \sim \vec{v}^{(i)}$, the randomized function $\mathcal{SM}$ satisfies $(\varepsilon, \max(\vec{v}, \vec{v}^{(i)}))$-differential privacy for both the privacy vector and the items. The maximum of the two vectors is taken point-wise. That is, if $\vec{v}' = \max(\vec{v}, \vec{v}^{(i)})$, then $v'_j = v_j = v_j^{(i)}$ for $j \neq i$ and $v'_i = \max(v_i, v_i^{(i)})$.

The privacy vector can thus be considered to be hidden and protected by the guarantees of heterogeneous differential privacy.

Theorem 2 (Protecting the privacy vector and items simultaneously)**.** *The randomized function $\mathcal{SM}$ provides ε-differential privacy for each individual privacy weight of $\vec{v}$ even if it changes simultaneously with its corresponding item in $\vec{d}$. This means that for all i, for all neighboring privacy vectors $\vec{v} \sim \vec{v}^{(i)}$, for all neighboring profiles $\vec{d} \sim \vec{d}^{(i)}$, for all outputs $t \in \mathbb{R}$, the following statement holds:*

$$\Pr[\mathcal{SM}(\vec{d}, \vec{v}, \varepsilon) = t] \leqslant \exp(\varepsilon \max(\vec{v}_i, \vec{v}_i^{(i)})) \Pr[\mathcal{SM}(\vec{d}^{(i)}, \vec{v}^{(i)}, \varepsilon) = t] \ . \tag{8}$$

Proof 4. *Let $\vec{d}_* = T(\vec{v}) \cdot \vec{d}$ and $\vec{d}_*^{(i)} = T(\vec{v}^{(i)}) \cdot \vec{d}^{(i)}$. Observe that $\vec{d}_*$ and $\vec{d}_*^{(i)}$ are neighbors on item i, since for $\vec{w} = T(\vec{v})$ and $\vec{w}^{(i)} = T(\vec{v}^{(i)})$ we have that $w_j d_j = w_j^{(i)} d_j^{(i)}$ for $j \neq i$. Moreover due to Proposition 1, they still belong to D. Consider*

$$\begin{aligned}
|f(\vec{d}_*) - f(\vec{d}_*^{(i)})| &= |f(T(\vec{v}) \cdot \vec{d}) - f(T(\vec{v}^{(i)}) \cdot \vec{d}^{(i)})| \\
&= |f(\mathrm{diag}(\vec{w}) \cdot \vec{d}) - f(\mathrm{diag}(\vec{w}^{(i)}) \cdot \vec{d}^{(i)})| \\
&= |f(\mathrm{diag}(\vec{w}^{(i)}) \cdot \vec{x}) - f(\mathrm{diag}(\vec{w}^{(i)}) \cdot \vec{d}^{(i)})| \\
&= |f(T(\vec{v}^{(i)}) \cdot \vec{x}) - f(T(\vec{v}^{(i)}) \cdot \vec{d}^{(i)})| \\
&\leqslant S_i(R(f, \vec{v}^{(i)})) \leqslant \vec{v}_i^{(i)} S(f)
\end{aligned}$$

in which $\vec{x}' = (d_1, \ldots, x_i, \ldots, d_n)$, in which $x_i = d_i w_i / w_i^{(i)}$. By symmetry we can show that

$$|f(\vec{d}_*) - f(\vec{d}_*^{(i)})| \leqslant \max(\vec{v}_i, \vec{v}_i^{(i)}) S(f) \ .$$

Therefore, we have:

$$\begin{aligned}
\frac{\Pr[\mathcal{SM}(\vec{d}, \vec{v}, \varepsilon) = t]}{\Pr[\mathcal{SM}(\vec{d}^{(i)}, \vec{v}^{(i)}, \varepsilon) = t]} &= \frac{h(t - R(f, \vec{v})(\vec{d}))}{h(t - R(f, \vec{v}^{(i)})(\vec{d}^{(i)}))} \\
&\leqslant \exp\left(\frac{\varepsilon |R(f, \vec{v})(\vec{d}) - R(f, \vec{v}^{(i)})(\vec{d}^{(i)})|}{S(f)}\right) \\
&= \exp\left(\frac{\varepsilon |f(\vec{d}_*) - f(\vec{d}_*^{(i)})|}{S(f)}\right) \\
&\leqslant \exp\left(\frac{\varepsilon \max(\vec{v}_i, \vec{v}_i^{(i)}) \cancel{S(f)}}{\cancel{S(f)}}\right) = \exp(\varepsilon \max(\vec{v}_i, \vec{v}_i^{(i)})),
\end{aligned}$$

in which $h(\cdot)$ is defined in Equation (4), thus proving the result. □

Corollary 1 (Protecting the privacy vector with $(\varepsilon, \max(\vec{v}, \vec{v}^{(i)}))$-DP)**.** *The randomized function $\mathcal{SM}$ provides ε-differential privacy for each individual privacy weight of $\vec{v}$. This means that for all neighboring privacy vectors $\vec{v} \sim \vec{v}^{(i)}$, for all outputs $t \in \mathbb{R}$ and profiles $\vec{d}$, the following statement holds:*

$$\Pr[\mathcal{SM}(\vec{d}, \vec{v}, \varepsilon) = t] \leqslant \exp(\varepsilon \max(\vec{v}_i, \vec{v}_i^{(i)})) \Pr[\mathcal{SM}(\vec{d}, \vec{v}^{(i)}, \varepsilon) = t] \ . \tag{9}$$

Proof 5. *Identical to the proof of Theorem 2.* □

Remark 2. *If the default privacy weight is set to 1 (*i.e.*, if this is the privacy weight automatically assigned to an item if the user did not manually specify it), then the resulting privacy guarantee if the user changes the item and its privacy weight simultaneously is ε-differentially privacy. To solve this issue and to obtain heterogeneous differential privacy, the default privacy weight could be set to zero. Alternatively, the privacy vector should be held constant and not being modified simultaneously with corresponding items.*

3.2.4 Utility of the Stretching Mechanism

Theorem 3 bounds the error introduced by the Stretching Mechanism. This theorem assumes that the function to be computed is K-Lipschitz Continuous (*cf.* Definition 9). However, it is possible to extend it to any function whose gradient is defined (*cf.* Lemma 3).

Definition 9 (K-Lipschitz continuous function)**.** *A function $f : \mathbb{R}^n \to \mathbb{R}$ is K-Lipschitz continuous, if for all $\vec{x}, \vec{y} \in \mathrm{Dom}(f)$:*

$$\frac{|f(\vec{x}) - f(\vec{y})|}{\|\vec{x} - \vec{y}\|} \leqslant K \ . \tag{10}$$

Remark 3. *The global sensitivity of a function is Dwork et al. (2006, Definition 2)*

$$S(f) = \sup_{x,y} \frac{|f(x) - f(y)|}{\|x - y\|_H} \ ,$$

in which $\|x - y\|_H$ is the Hamming distance between the two vectors x and y. For binary vectors (those belonging to $\{0,1\}^n$), the Hamming distance coincides with the ℓ_1 distance, which means that $\|x-y\|_H = \|x-y\|_1$. Therefore, since $\|x-y\| \leqslant \|x-y\|_1 = \|x-y\|_H$, we can observe that:

$$K = \sup_{x,y} \frac{|f(x) - f(y)|}{\|x - y\|} \geqslant \sup_{x,y} \frac{|f(x) - f(y)|}{\|x - y\|_H} = S(f) \ .$$

As a consequence for binary vectors, K-Lipschitz continuous functions with small K have low global sensitivity.

Theorem 3 (Utility theorem)**.** *Let $f : D \to \mathbb{R}$ be a K-Lipschitz continuous function from a semi-balanced set D to the reals, and let $\vec{v} \in [0,1]^n$ be a privacy vector and $T : [0,1]^n \to \mathbb{R}^{n \times n}$ be a function taking a privacy vector to a shrinkage matrix. Finally, let R be a mapping sending a function f and a privacy vector $\vec{v}$ to the function $R(f, \vec{v}) : D \to \mathbb{R}$ such that $R(f, \vec{v})(\vec{d}) = f(T(\vec{v}) \cdot \vec{d})$ for all vectors $\vec{d}$.*

If $\mathcal{SM}(\vec{d}) = R(f, \vec{v})(\vec{d}) + \mathrm{Laplace}(0, S(f)/\varepsilon)$ is the Stretching Mechanism, then

$$\Pr[|\mathcal{SM}(\vec{d}) - f(\vec{d})| > k] \leqslant \frac{1}{2} \exp\left(\frac{\varepsilon(K(1 - \underline{w})\|\vec{d}\| - k)}{S(f)}\right) \ , \tag{11}$$

in which $\underline{w}$ is the smallest value in the shrinkage matrix $T(\vec{v})$ and the probability is taken over the added Laplacian noise.

Corollary 2 (Utility theorem for inner product computed on binary vectors)**.** *If f is the inner product function and $\vec{d} \in \{0,1\}^n$, then*

$$\Pr[|\mathcal{SM}(\vec{d}) - f(\vec{d})| > \sqrt{n}] \leqslant \frac{1}{2}\exp(-\varepsilon \underline{v}\sqrt{n}) \ , \tag{12}$$

in which $\underline{v}$ is the smallest privacy weight.

Corollary 2 holds since $\|\vec{d}\| \leqslant \sqrt{n}$ for binary vectors, and thus for the inner product of binary vectors $K = S(f) = 1$ and $v = w$.

In the rest of this section we provide the proof of Theorem 3. The following lemma is needed to prove this theorem for K-Lipschitz continuous functions. The alternative lemma for any function whose gradient is defined is Lemma 3.

Lemma 2 (Bounding the bias for K-Lipschitz continuous functions)**.** *If f is K-Lipschitz continuous then for all $\vec{d} \in \mathrm{Dom}(f)$ and all $\vec{w} \in [0,1]^n$, then*

$$|f(\vec{d}) - f(\mathrm{diag}(\vec{w}) \cdot \vec{d})| \leqslant K(1 - \underline{w})\|\vec{d}\| \ , \tag{13}$$

in which $\underline{w}$ is the smallest value among $w_1, \ldots, w_n$.

Proof 6. *We have*

$$|f(\vec{d}) - f(\mathrm{diag}(\vec{w}) \cdot \vec{d})| \leqslant K\|\vec{d} - \mathrm{diag}(\vec{w}) \cdot \vec{d}\| = K\|M\vec{d}\| \leqslant K\|M\|\|\vec{d}\| \ , \tag{14}$$

in which $M = I - \mathrm{diag}(\vec{w})$, and $\|M\|$ is the matrix norm. The first inequality follows because f is K-Lipschitz continuous. The second inequality follows from the definition of the matrix norm[3]*.*

Since for the ℓ_2 norm, $\|M\|$ is the spectral norm of M, which equals $\sigma_{\max}(M)$, the largest singular value of M, and since M is a diagonal matrix whose entries are in $[0,1]$, then $\|M\| = \sigma_{\max}(M) = 1 - \underline{w}$. □

Proof 7 (Proof of theorem 3)**.** *Let N be a random variable drawn from* Laplace$(0, b)$*. Then for any positive k,*

$$\Pr[|\mathcal{SM}(\vec{d}) - f(\vec{d})| > k] = \Pr[|N + f(T(v) \cdot \vec{d}) - f(\vec{d})| > k].$$

[3]The matrix norm is defined as $\|A\| = \sup_{x \neq 0} \frac{\|Ax\|}{\|x\|}$. It directly implies that for any non-zero x, $\frac{\|Ax\|}{\|x\|} \leqslant \|A\|$, *i.e.* $\|Ax\| \leqslant \|A\|\|x\|$.

Let $r = f(T(v)\cdot\vec{d}) - f(\vec{d})$. *Then,*

$$\begin{aligned}\Pr[|N+r|>k] &= 1-\Pr[|N+r|\leqslant k] = 1-\begin{cases}\exp(-|r|/b)\sinh(k/b) & \text{if } |r|\geqslant k\\ 1-\exp(-k/b)\cosh(r/b) & \text{otherwise}\end{cases}\\ &= \begin{cases}1-\exp(-|r|/b)\sinh(k/b) & \text{if } |r|\geqslant k\\ \exp(-k/b)\cosh(|r|/b) & \text{otherwise}\end{cases} \leqslant \exp(-k/b)\cosh(|r|/b)\\ &= \frac{1}{2}(\exp(\frac{|r|-k}{b})+\exp(-\frac{|r|+k}{b})) \leqslant \frac{1}{2}\exp(\frac{|r|-k}{b})\\ &\leqslant \frac{1}{2}\exp(\frac{\varepsilon(K(1-\underline{w})\|d\|-k)}{S(f)}).\end{aligned}$$

The first inequality follows because for nonnegative x *and* y, $1\leqslant\cosh(x-y) = (\exp(x-y)+\exp(y-x))/2 = \exp(-y)\cosh(x)+\exp(-x)\sinh(y)$, *and thus* $\exp(-y)\cosh(x)\geqslant 1-\exp(-x)\sinh(y)$. *The last inequality follows from the bound on* $|r|$ *from Lemma 2 and the fact that* $b = S(f)/\varepsilon$ *from the Laplacian Mechanism.* □

In the last inequality, the bound from Lemma 3 can be substituted to obtain a utility theorem for any function whose gradient is defined as we describe in the following.

Lemma 3 (Bound on the bias for functions whose gradient is defined)**.** *If* $f:\mathbb{R}^n\to\mathbb{R}$ *is a function whose gradient is defined then for all* $\vec{d}\in\mathrm{Dom}(f)$ *and all* $\vec{w}\in[0,1]^n$, *then*

$$|f(\vec{d})-f(\mathrm{diag}(\vec{w})\cdot\vec{d})| \leqslant (1-\underline{w})\|\vec{d}\|\max_{0\leqslant c\leqslant 1}\|\nabla f(B\cdot\vec{d})\| \ , \tag{15}$$

in which $B = cI+(1-c)\,\mathrm{diag}(\vec{w})$, $\underline{w} = \min_i w_i$, *and* ∇f *is the gradient of the function* f.

Proof 8. *Let* $\vec{y} = \vec{d}$ *and* $\vec{x} = \mathrm{diag}(\vec{w})\cdot\vec{d}$, *then by the mean value theorem (Dym, 2007, Theorem 14.4, p. 301), there exists a constant* $0\leqslant c\leqslant 1$ *(depending on* $\vec{d},\vec{w}$, *and* f*) such that* $f(\vec{y})-f(\vec{x}) = \nabla f((1-c)\vec{x}+c\vec{y})\cdot(\vec{y}-\vec{x})$, *in which* $\cdot$ *denotes the scalar product. Therefore, by the Cauchy-Schwarz inequality, we have* $|f(\vec{y})-f(\vec{x})|\leqslant\|\nabla f((1-c)\vec{x}+c\vec{y})\|\|\vec{y}-\vec{x}\|$. *Finally, the lemma follows by observing that* $\|\vec{y}-\vec{x}\| = \|\vec{d}-\mathrm{diag}(\vec{w})\cdot\vec{d}\| = \|(I-\mathrm{diag}(\vec{w}))\cdot\vec{d}\|\leqslant\|I-\mathrm{diag}(\vec{w})\|\|\vec{d}\| = (1-\underline{w})\|\vec{d}\|$, *in which* $\underline{w}$ *is the minimum of* $\vec{w}$, *and* $(1-c)\vec{x}+c\vec{y} = (1-c)\,\mathrm{diag}(\vec{w})\cdot\vec{d}+c\vec{d} = (cI+(1-c)\,\mathrm{diag}(\vec{w}))\cdot\vec{d} = B\cdot\vec{d}$. □

Note that if the norm of the gradient of the function f is bounded from above by a constant, we could apply instead Lemma 2.

For inner product function on binary vectors $f(\vec{d}) = \sum_{i=1}^{n/2} d_i d_{i+n/2}$, the gradient is $\|B\cdot\vec{d}\|\leqslant\|\vec{d}\|$ (since B is a shrinkage matrix). Hence, the bound on the bias will be $(1-\underline{w})\|\vec{d}\|^2$. Moreover, since for the inner product function $\vec{w} = \vec{v}$, the bias will be less than $(1-\underline{v})n$. From this, we can constrain $\underline{v}$ to guarantee any desired upper bound on the bias. For instance, if the bias should be $\widetilde{O}(\sqrt{n})$, then $\underline{v} = \widetilde{\Omega}(1/\sqrt{n})$.

4 HDP in Practice

To assess the practicality of our approach, we have applied the HDP mechanism on a collaborative social system (Bertier et al., 2010), and evaluated its impact on a related semantic clustering task. In this collaborative social system, each user (*i.e.* node) is associated with a profile. A profile is the set of items the user has liked or tagged (*e.g.*, the set of URLs in his Delicious account). The objective of the semantic clustering task is to assign each node with the k-closest neighbors according to a given similarity metric. In this paper, we use the classical *cosine similarity* (introduced later) to quantify the similarity between two profiles. The task is carried out using a fully distributed protocol, therefore the nodes compute locally (*i.e.*, without relying on a central authority) their similarity with other profiles.

4.1 Applying HDP to Semantic Clustering

In the context of distributed semantic clustering, we are interested in providing heterogeneous differential privacy guarantees to the profiles of nodes (*i.e.*, users). More precisely, we consider the scenario in which a particular user can assign a privacy weight, between 0 and 1, to each item of his profile. The value 0 corresponds to the strongest privacy guarantee in the sense that the presence (or absence) of this item will not affect the outcome (the clustering) at all, while the value 1 is the *lowest* level of privacy possible in our framework (however it still provides the standard guarantees of ε-differential privacy). Thus, the privacy weights of a user directly reflect his privacy attitudes with respect to particular items of his profile, and as a side effect determines the influence of this item in the clustering process. In particular, an item with a higher weight will contribute more to the clustering process, while a item with a lower weight will influence less the resulting clustering.

The *cosine similarity* between two profiles X and Y is defined as

$$\mathsf{cos\_sim}(X,Y) = \frac{|X \cap Y|}{\sqrt{|X| \times |Y|}} \ , \tag{16}$$

such that $|X \cap Y|$ is the number of items in common between X and Y, and $|X|$ and $|Y|$ correspond to the number of items of X and Y, respectively.

The *indicator function* of a profile, when it is represented as a binary vector, for the item i is 1 if the i^{th} item is present in the profile and 0 otherwise. More formally, the i^{th} coordinate $\chi_i(x)$ of the indicator function $\chi(x)$ of the profile x is denoted by:

$$\chi_i(x) = \begin{cases} 1 & \text{if } i \in x \\ 0 & \text{otherwise} \end{cases} \ .$$

Using the notation of the indicator function, the cosine similarity could be defined as

$$\frac{\chi(X) \cdot \chi(Y)}{\|\chi(X)\|_2 \|\chi(Y)\|_2} \ ,$$

in which the operation "$\cdot$" denotes the scalar product. In the following, we apply HDP to the scalar product function and use this modified version to compute the cosine similarity on profiles represented as binary vectors.

Given two profiles X and Y and their corresponding indicator functions $\vec{x} = \chi(X)$ and $\vec{y} = \chi(Y)$, let $\mathsf{SP}(\vec{x}, \vec{y}) = \sum_i x_i y_i$ refers to the scalar product between the two profiles. The privacy vector $\vec{v}$ is composed of two parts, one for the profile $\vec{x}$ and the other for the profile $\vec{y}$: $(\vec{v^x}, \vec{v^y})$. Consider the matrix $T(\vec{v}) = \mathrm{diag}(v)$ and let $R(\mathsf{SP}, \vec{v}) = \mathsf{SP}(T(\vec{v^x}) \cdot \vec{x}, T(\vec{v^y}) \cdot \vec{y})$ be the Stretching Mechanism, in which T is the stretch specifier. This mechanism R satisfies the premise of Theorem 1 and therefore the choice of $T(\vec{v}) = \mathrm{diag}(\vec{v})$ also ensures HDP, as proven in the following lemma.

Lemma 4. *Consider a matrix $T(\vec{v}) = \mathrm{diag}(\vec{v})$ and a mechanism $R(\mathsf{SP}, \vec{v}) = \mathsf{SP}(T(\vec{v^x}) \cdot \vec{x}, T(\vec{v^y}) \cdot \vec{y})$, such that $\vec{x}$ and $\vec{y}$ correspond to profiles and $v^{\vec{x}}$ and $v^{\vec{y}}$ to their associated privacy vectors. In this situation, the following statement is always true: $S_i(R(\mathsf{SP}, \vec{v})) \leqslant v_i S(\mathsf{SP})$ for all i.*

Proof 9. *Each profile being represented as a binary vector, the global sensitivity of the scalar product is one (*i.e.*, $S(\mathsf{SP}) = 1$). Thereafter, for the sake of simplicity, let R denotes $R(\mathsf{SP}, \vec{v})$. As $T(\vec{v})$ is a diagonal matrix, it is strictly identical to its transpose $T(\vec{v})^\top$. We can assume without loss of generality that $\vec{d}^{(i)} = (\vec{x}, \vec{y}^{(j)})$ for item $j = i - \dim(x)$, and therefore that:*

$$\begin{aligned} S_i(R) &= \max_{\vec{d} \sim \vec{d}^{(i)}} |T(\vec{v^x})\vec{x} \cdot T(\vec{v^y})\vec{y} - T(\vec{v^x})\vec{x} \cdot T(\vec{v^y})\vec{y}^{(j)}| \\ &= \max_{\vec{d} \sim \vec{d}^{(i)}} |(\vec{x}^\top T(\vec{v^x}) T(\vec{v^y})) \cdot (\vec{y} - \vec{y}^{(j)})|. \end{aligned}$$

However the vector $\vec{y} - \vec{y}^{(j)}$ has all its coordinates set to 0 except for the j^{th} coordinate. Therefore, the maximum is reached when $y_j = 1$, $\vec{x} = \vec{1} = (1, \cdots, 1)$, and is such that:

$$\vec{v^x_j} \vec{v^y_j} \leqslant v_i = v_i \times 1 = v_i S(\mathsf{SP}),$$

which concludes the proof. □

The previous lemma proves that the proposed modified version of scalar product is differentially private, while the next lemma simply states that if we rely on this differentially private version of scalar product to compute the cosine similarity (or any similar metric), the outcome of this computation will still be differentially private. A standard (*i.e.*, non-heterogeneous) version of the following post-processing lemma can be found in the literature (Kasiviswanathan et al., 2008), which we have generalized to heterogeneous differential privacy.

Lemma 5 (Effect of post-processing on HDP)**.** *If a randomized function $\hat{f}$ satisfies $(\varepsilon, \vec{v})$-differential privacy, then for any randomized function $g : \mathsf{Range}(\hat{f}) \to \mathbb{R}$ independent of the input, the composed function $g \circ \hat{f}$ satisfies also $(\varepsilon, \vec{v})$-differential privacy. The randomness of the function g is assumed to be independent of the randomness of $\hat{f}$ in order for this property to hold.*

Proof 10. *The theorem is equivalent to prove that for any two neighboring profiles* $\vec{d} \sim \vec{d}^{(i)}$ *the following holds:*

$$\Pr[g \circ \hat{f}(\vec{d}) = t] \leqslant \exp(\varepsilon v_i) \Pr[g \circ \hat{f}(\vec{d}^{(i)}) = t].$$

To prove this, consider any two neighboring profiles $\vec{d} \sim \vec{d}^{(i)}$*:*

$$\begin{aligned}
\Pr[g \circ \hat{f}(\vec{d}) = t] &= \int_{s \in \mathsf{Range}(\hat{f})} \Pr[\hat{f}(\vec{d}) = s] \cdot \Pr[g(s) = t] \\
&\leqslant \int_{s \in \mathsf{Range}(\hat{f})} \exp(\varepsilon v_i) \Pr[\hat{f}(\vec{d}^{(i)}) = s] \cdot \Pr[g(s) = t] \\
&= \exp(\varepsilon v_i) \Pr[g \circ \hat{f}(\vec{d}^{(i)}) = t],
\end{aligned}$$

thus concluding the proof. □

4.2 Experimental Evaluation

For the experiments, we assume that in reality, nodes will assign different privacy weights to the items in their profiles. In order to simulate this, we generate privacy weights uniformly at random from a set of n equally-spaced values in a fixed range $[\underline{u}, \overline{u}]$. More formally, each item is associated with a privacy weight sampled uniformly at random from the set $\{\underline{u}, \overline{u}+\delta, \ldots, \overline{u}-\delta, \overline{u}\}$, $\delta = (\overline{u}-\underline{u})/(n-1)$, for $0 \leqslant \underline{u} < \overline{u} \leqslant 1$. For instance, if $\underline{u} = 0.5$, $\overline{u} = 1$ and $n = 3$, then the weights assigned to items will be uniformly chosen from the set $\{0.5, 0.75, 1\}$.

We run our experiments on three datasets coming respectively from Delicious, Digg and a survey conducted within our lab. About 113 users participated in the survey and submitted their feedback (in forms of like/dislike) on 196 pieces of news. Therefore, in the survey dataset a user's profile consists of the news he has liked, while for the Digg dataset a profile consists of the items that a user has forwarded to others users. Finally, in the Delicious dataset, the profile of the user consists of the items he has tagged.

- *Delicious dataset.* Delicious (`delicious.com`) is a collaborative platform for keeping bookmarks in which users can tag the URLs of websites they liked. The Delicious dataset consists in the profiles of 504 users, a profile being a set of URLs that the user has tagged. The total number of URLs in the collective set of users' profiles is 51,807 URLs. In such a setting, the problem of similarity computation arises naturally, when providing personalized services such as the recommendation of URLs drawn from the ones tagged in Delicious. For the sake of simplicity, in the experiments conducted, each URL was assigned a unique identifier in the range of $\{1, \ldots, 51807\}$, in order to handle identifiers as integers instead of URL strings. The average size of a profile is 135 URLs, indicating that this dataset is sparse.

- *Digg dataset.* The dataset consists of 481 users of Digg (digg.com), a social news website. The profile of these users is composed of the news that they have shared over a period of 3 weeks in 2010. All the users considered have shared more than 7 items per week and the dataset contains 1237 items, each of which has been shared by at least 10 users. The average size of a profile is 317 items, indicating that this dataset is dense.

- *Survey dataset.* Around 196 randomly chosen pieces of news on various topics have been shown to 113 colleagues and relatives, who have then submitted their opinion in terms of like/dislike for each news. The average size of the profile is 68. Indeed, while each user has answered to all the 196 pieces of news, he has only liked 68 of those pieces of news on average.

The distributed clustering algorithm is gossip-based and works in an iterative manner (Bertier et al., 2010). To assess the quality (*i.e.*, utility) of a particular clustering, we rely on the *recall* metric. The recall can be defined as the ratio between the number of search items a node could find in the collective profiles of his k closest neighbors (as induced by the clustering) over all possible items of his profile. We consider this metric for our experiments but other standard metrics used in recommendation systems could be used as well. In the experiments conducted, the profile of each user is split at random into a training set composed of 90% of the profile while the remaining 10% is used for testing. The items selected for testing must be in the profiles of at least two nodes. After 20 rounds of exchanging gossip messages during the clustering protocol, each user searches for those 10% of items in the profiles of the k closest neighbors provided by the clustering protocol (in all our experiments, $k = 10$). In this situation, the recall is equal to the ratio of items found in the collective profiles of the neighbors over all the possible items contained in the testing set. The average recall of all users is then reported as the outcome of the experiment. In all the following experiments, we set $\varepsilon \in \{0.1, 0.5, 1, 2, 3\}$, and the result is averaged among these values. The source code of our experiments (but not the datasets) is available publicly at https://github.com/malaggan/heterogeneous-differential-privacy.

In Figure 1, we have plotted the three cases for which the interval $(\underline{u}, \overline{u})$ is set to be $(0, 1), (0.5, 1)$, and $(0.9, 1)$. The x-axis represents $\underline{u}$, while the y-axis is the recall averaged over all slices (from $n = 1$ to $n = 10$) for the experiment in the range $[x, 1]$. Afterwards in Figure 2, we have fixed the range $\underline{u} \in \{0, 0.5, 0.9\}$ and $\overline{u} = 1$ and plot the average recall over all users over all runs versus n, the number of slices (ranging from 1 to 10). In both figures, the error bars represent 99% confidence interval around the mean.

From Figure 1 (Survey and Digg), we can observe that there is not much difference in terms of utility between the situations in which $\underline{u} = 0.5$ and $\underline{u} = 0.9$, as both situations are close to the utility obtained with the baseline algorithm. Indeed, the largest difference is obtained when $\underline{u}$ is set to 0, in which case the utility gets closer to the utility obtained through a random clustering. Furthermore, Figure 2 (Survey and

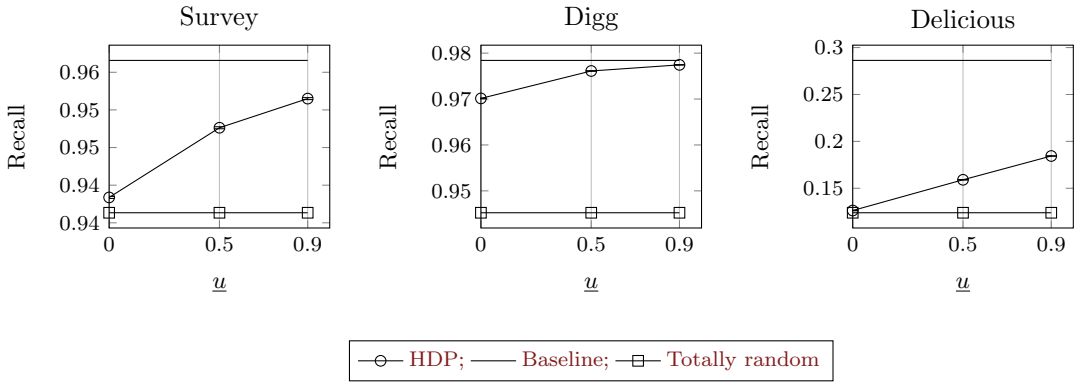

Figure 1: The value reported is the average recall obtained when all peers have the same distribution over privacy weights for all items, averaged over the number of slices. *Baseline* refers to the recall obtained when the system run with no privacy guarantees using the plain version of the clustering algorithm, while *Random* refers to a random clustering process in which peers choose their neighbors totally at random. Bars represent 99% confidence interval around the mean (they are almost invisible for Digg and Delicious).

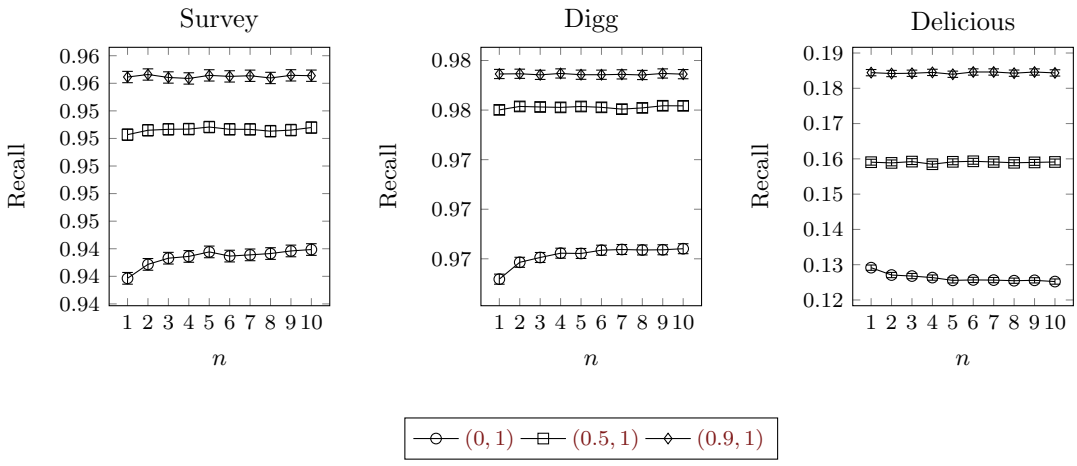

Figure 2: The value reported is the average recall obtained when all peers have the same distribution over privacy weights for all items, plotted against the number of slices. Bars represent 99% confidence interval around the mean.

Digg) demonstrates that varying the number of slices has almost no effect on the utility achieved by $\underline{u} \in \{0.5, 0.9\}$, but has significant impact on the situation in which $\underline{u} = 0$, for which the utility decreases with a wider gap. We can notice that the number of slices have small to no effect.

4.3 Varying Privacy Attitudes Among Users

The results of the previous section were obtained for the setting in which all nodes draw their privacy weights from the same distribution (*i.e.*, all users have the same privacy attitude). However, according to a survey (Jensen et al., 2005), users of information systems can be classified in at least three very different groups called the *Westin categories* (Harris Interactive, 2003). These three groups are: Privacy Fundamentalists, Privacy Pragmatists and Privacy Unconcerned. The first group is composed of the users concerned about their privacy, while on the contrary the third group is composed of the ones that are the least concerned (according to a particular definition of concern detailed in the cited poll) and finally the second group is anything in between. For the following experiments, we have adopted the spirit of this classification and consider the three groups of users defined thereafter.

Each group is equipped with a different distribution from which they pick their privacy weights as follows.

1. The Unconcerned group corresponds to users that do not really care about their privacy and thus all their items have a privacy weight of 1.

2. The Pragmatists group represent users that care a little bit about their privacy, such that all their items have a privacy weight chosen uniformly at random among $\{0.5, 0.75, 1\}$.

3. The Fundamentalists group embodies users that really care a lot about their privacy and whose items have a privacy weight chosen uniformly at random among $\{0, 0.5, 1\}$.

The main issue we want to investigate is how the presence of a relatively conservative group (*i.e.*, having relatively high privacy attitudes) affect the utility of other groups. More specifically, we want to measure whether or not the presence of a group of nodes with high privacy attitudes indirectly *punish* (*i.e.*, reduce the utility) of other more open groups.

During the experimentations, we have tried different proportions of these groups for a total number of users of 500. Each value plotted in Figure 3, has been averaged over 10 runs (in each run a random 10% of the users are removed) but the partition in groups is fixed for a given set of runs. All experiments are averaged on $\varepsilon \in \{0.1, 0.5, 1, 2, 3\}$. According to a 2004 poll (Jensen et al., 2005), the percentage of each of the privacy groups Fundamentalists, Pragmatists and Unconcerned are respectively, 34%,43% and 23%. Nonetheless, we also experiment a combination of several other distributions in

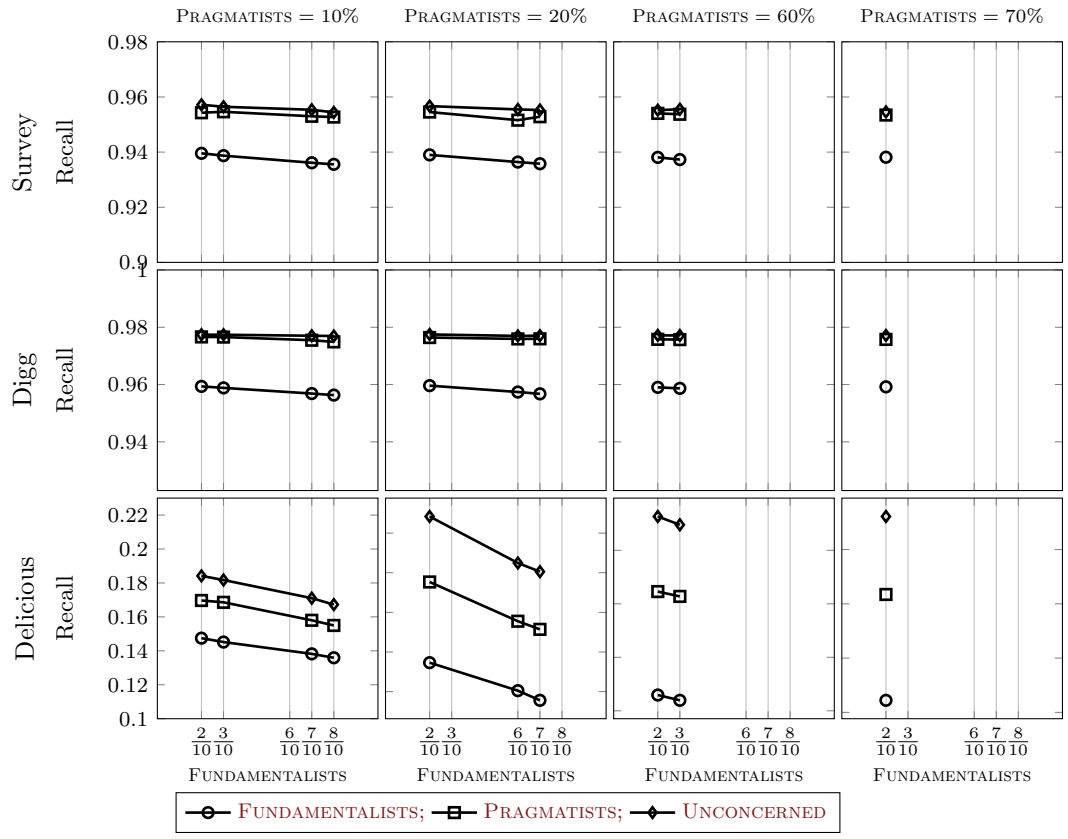

Figure 3: Results obtained for the Delicious, Digg and survey datasets. The heterogeneous differential privacy has been computed for 3 groups with different privacy attitudes. For a particular figure and a particular x tick, the percentage of Unconcerned group is fully determined as (1− Pragmatists − x).

order to investigate other possible settings. In particular, we have also tried the following percentages for each group: the proportion of the Unconcerned group and Pragmatists group vary in the following range $\{10\%, 20\%, 60\%, 70\%\}$, while the Fundamentalists group is assigned to the remaining percentage (*i.e.*, there is only two degrees of freedom). If Unconcerned group + Pragmatists group $> 100\%$, then this combination is discarded. In Figure 3, the x-axis represents the percentage of the Fundamentalists group, while the y-axis corresponds to the recall. Each of the three lines correspond to the recall of one of the three groups (Fundamentalists, Pragmatists, and Unconcerned). For each of the four plots, the proportion of the Pragmatists group is denoted in the plot by the expression *Pragmatists = some value*. The proportion of the remaining group (Unconcerned) can be directly inferred by subtracting

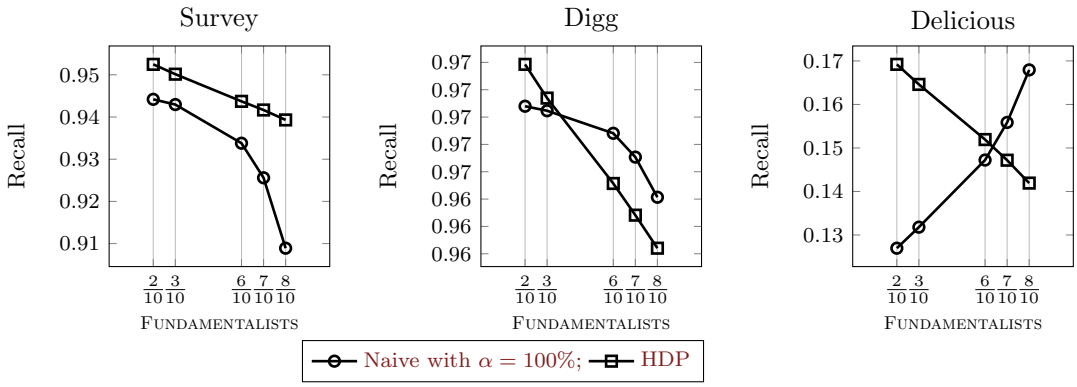

Figure 4: The average recall obtained for the Stretching Mechanism versus the Naïve baseline in which 100% of the Fundamentalists are removed and the remaining users are given homogeneous privacy guarantee equivalent to the preference of Pragmatists.

the proportions of the two other groups from 100%.

From the results obtained, we can conclude that (1) Pragmatists and Unconcerned always have better recall than Fundamentalists and (2) Unconcerned often have a better recall than Pragmatists, though for some datasets the difference appears to be negligible. This seems to indicate that the group caring more about privacy usually is punished more (*i.e.*, its utility is lower) than groups that are more liberal with respect to privacy expectations. This not really surprising as a low privacy weight will result in users from the Fundamentalists group segregating themselves from other users in the clustering to the point that they will not necessarily have meaningful neighbors in their view. Finally, to the question whether (or not) more liberal groups will be punished by conservative groups, the answer seems to be negative. Indeed it can be seen from the results of the experiments, that conservative groups are punished more than liberal groups. For instance, the utility of liberal groups only decreases from 0.22 to 0.19 as the percentage of conservative groups increases from 20% to 80%.

In order to test whether heterogeneous differential privacy (HDP) can give better utility than homogeneous differential privacy, we compare our experiments to the naïve alternative scenario (we call it "Naïve" hereafter) in which all the Fundamentalists are removed from the dataset and then both the Pragmatists and Unconcerned receive homogeneous privacy guarantees strong enough to match the relatively strict privacy preference of Pragmatists. That is, the privacy weights of Pragmatists and Unconcerned are set to 0.5 (the strictest privacy weight a Pragmatist can choose), effectively rendering the case identical to the homogeneous $\varepsilon/2$-differentially private case. In the experiment, we actually remove only $\alpha \in \{0\%, 20\%, 40\%, 60\%, 80\%, 100\%\}$ fraction of the Fundamentalists to observe how their removal impacts utility. In the situations in which some of the Fundamentalists are retained, they also receive a

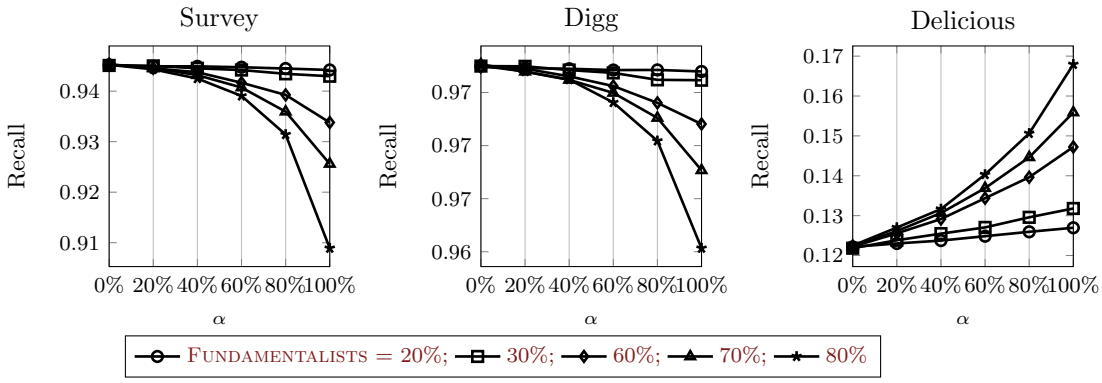

Figure 5: The average recall obtained by the Naïve baseline in which α fraction of the Fundamentalists are removed and the remaining users (including the leftover Fundamentalists) are given homogeneous privacy guarantee equivalent to the preference of Pragmatists.

privacy weigh of 0.5 to all their items, since this is a homogeneous differential privacy experiment. However, this means that their privacy is not protected as they would expect and that the utility they receive as well as the overall utility of the system is higher than it should be given their (overridden) privacy choice. The results are presented in Figure 4 and Figure 5. We observe from Figure 4 that HDP outperforms Naïve for all Fundamentalists ratios for the Survey dataset.

However, a more interesting pattern emerges for the two other datasets. For Digg and Delicious, HDP outperforms Naïve when the Fundamentalists ratio is respectively less than 40% and 70%. This shows that when the number of Fundamentalists is too big, it is better to throw them out and provide homogeneous privacy than to keep them and accommodate for heterogeneous privacy needs. This may be due to fact that Fundamentalists add noise to the network. Some nodes will choose them because of the high noise, over other nodes who are actually more similar to them. The problem with that is that a node keeps only a fixed number of close neighbors (10 in our experiments), and if some of these slots are wasted by Fundamentalists who are mistaken for similar nodes, then fewer slots are available for the actual similar nodes. An additional reason for this behavior is that when the Fundamentalists are thrown away, the remaining nodes spend less time (cycles) to find similar nodes, since they do not waste cycles on the removed Fundamentalists. A node typically explores between 1 and 3 other nodes in each cycle. The number of cycles in our experiments is 20, and if some of these cycles are wasted to test Fundamentalists for similarity then less cycles are spent on testing other nodes which are more likely to provide useful similarity information. The overall utility depends crucially on the performance of the distributed clustering algorithm, which in itself depends crucially on maximizing the useful information resulting from each cycle and each similarity computation. A majority of Fundamentalists overwhelming each cycle with almost

useless similarity computation circumvents the clustering algorithm, thus removing them would be a better option. This intuition is supported by Figure 5, in which we can observe that for Delicious, the utility consistently enhances as more FUNDAMENTALISTS are removed from the system (*i.e.*, as α increases). However, as this is not true for Digg, it suggests that the harmful effect of a majority of FUNDAMENTALISTS is amplified in sparse datasets, in which extracting more useful similarity information is more crucial than in denser datasets like Digg, in which it may be relatively straightforward.

5 Related Work

The majority of previous works on heterogeneous privacy has focused only on user-grained privacy (Das et al., 2011; Kumar et al., 2010), in which each user may define his own privacy level (instead of having the same privacy guarantee for all users across the system). As opposed to item-grained privacy, which allows each item of an individual user to have a different privacy weight, user-grained privacy restricts all the items of the same user to the same privacy weight. For instance, Das et al. (2011) have proposed a secure protocol for aggregating sums in a P2P network. In this setting, each node has an input vector, which could be, for instance, a profile. In this protocol, each node picks at random a few other nodes of the system with whom it computes some local function[4] in a private manner (the local function begins with a sum as well). The more peers a specific node chooses to participate to the computation, the higher the privacy will be obtained by this node according to the considered definition of privacy. More precisely in their setting, privacy is mainly quantified by the probability of collusion of the peers chosen by a particular node when the aggregation protocol is run. This probability can be made smaller by choosing a larger set of peers, the main intuition being that for a particular node running the aggregation protocol with a larger group diminishes the probability that all these nodes will collude against him. Thus, the best privacy guarantees could be obtained by running the protocol with the entire set of peers, but this would be too costly in practice. The main objective in this protocol is to be adaptive by providing a trade-off between the privacy level chosen by a user and the resulting cost in terms of computation and communication. In particular, each user has the possibility to choose heterogeneously the peers with whom he wants to run the aggregation protocol by taking into account his own privacy preferences. However, this work does not seem to be easily extendable to integrate item-grained privacy.

Another work due to Kumar et al. (2010) is a form of generalization of k-anonymity (Sweeney, 2002). The standard definition of k-anonymity requires that in the sanitized database that is released, the profile of a particular individual should be indistinguishable from at least $k-1$ other individuals (thus here k can be considered as being the privacy parameter). The proposed generalization (Kumar et al., 2010) essentially enables each user to require a different value for k for each attribute in his profile. For example, a user may require that his data should be included in the published database only if there are at least 4 other users sharing his ZIP code and at least 8 other users

[4]The function is local in the sense that it depends only on the inputs of the node and the peers it has chosen.

whose age difference with him is at most 3 years. The possibility of setting the range of a particular attribute could be regarded as item-grained heterogeneous privacy in the sense that an attribute whose privacy range is large is less likely to be useful for de-anonymizing the user than an attribute whose privacy range is less. To summarize, the main objective of this approach is to protect the privacy of a user by anonymizing it (*e.g.*, to prevent de-anonymization and linking attacks), while in our work the main objective is to prevent the possibility of inferring the presence or absence of a particular item in the profile.

A line of research on auctions *for* privacy has provided almost the same definition for the heterogeneous differential privacy as ours (Ghosh and Roth, 2011; Dandekar et al., 2011). The main difference with our contribution is that these previous works do not provide a mechanism to realize heterogeneous difference privacy, but instead only use the definition to achieve the post-release privacy guarantees. In the model studied, the participants are composed of a data analyst and a group of users. Each user has as input a private bit and the data analyst wants to estimate in a differentially-private manner a global function of the private bits of all users, such as the sum or the weighted sum. The data analyst is willing to pay each user for the loss of privacy he incurred by participating in this process. More precisely, each user i has a *privacy valuation* $v_i(\varepsilon_i) : \mathbb{R}_+ \to \mathbb{R}_+$ indicating the amount of his loss given the privacy guarantee he gets. The user has no control over ε_i (*i.e.*, the privacy guarantee he ends up with), which is decided solely by the auction mechanism. As such the valuation function v_i merely affects the payment of the user, as his payment is decided indirectly by the mechanism given the valuation function and is not decided directly by him. Therefore, our work is incomparable to theirs, because the privacy parameter ε_i acts mainly as an indication about the level of privacy reached, while in our setting the privacy parameter represents the *user's requirement* about the privacy of a particular item of his profile. In Ghosh and Roth (2011), users are divided into two sets. One of the set of users will not be included in the query, thus effectively having absolute privacy, while the remaining set of users end up having completely homogeneous privacy guarantees. More precisely, each user in the remaining set ends up having ε-differential privacy, with some ε being the same for all of these users. In contrast in Dandekar et al. (2011), users effectively have heterogeneous privacy guarantees. However, these guarantees are determined by the public weights of the auctioneer, which the auctioneer chooses so as to compute the weighted average of the users' inputs and independently of the privacy valuations of the users.

Nissim et al. (2007) have investigated how the amount of noise necessary to achieve differential privacy can be tailored by taking into account to the particular inputs (*i.e.*, profiles) of participants, in addition to the sensitivity of the function considered. The main objective of this approach is to reduce the amount of noise that needs to be added to inputs that are not *locally sensitive* (*i.e.*, for which the output does not change much if only one item is changed). However, they also show that the amount of noise added may itself reveal information about the inputs. Hence, they defined a differentially private version formalizing the notion of local sensitivity called *smooth sensitivity*, guaranteeing that the amount of noise added is itself ε-differentially private. Similarly, we have

ensured that for our notion of heterogeneous differential privacy, the amount of noise added is not impacted by the specific profile considered or by the privacy requirements formulated by a user. Rather, we have modified the function under consideration and its sensitivity, which also impacts the distortion induced of the output (*cf.*, Section 3.2.4). We have also proven that the privacy requirements of a user expressed in the form of *private weights* remain private as they are also covered by ε-differentially privacy guarantees. Thus, it is difficult for an adversary observing the output of an heterogeneous differentially private mechanism to guess the privacy weight that a user has put on a particular item of his profile.

In a recent work, Jorgensen et al. (2015) independently developed the concept of *Personalized Differential Privacy* (PDP), a notion of privacy similar to HDP but providing non-uniform user-grained (as opposed to item-grained) and *public* privacy weights. They also propose a Sampling mechanism to achieve PDP. The Sampling mechanism transforms any differentially private algorithm into a personally differentially private one by introducing a preprocessing step that non-uniformly samples each user's data with a probability proportional to his *public* privacy weight. The Sampling mechanism can support categorical attributes unlike our Stretching mechanism. They also design another mechanism based on the exponential mechanism that can provide PDP for functions like median and min/max that our Stretching mechanism does not support. For the use case of user-grained privacy considered in their work (Jorgensen et al., 2015), maintaining the secrecy of the privacy weight of each user is not crucial if the value of the user's privacy weight is based *only* on information not related to his data (*e.g.*, such as his social status). In contrast, we stress that for our use case of item-grained privacy, maintaining the secrecy of the privacy weights is very important as the privacy weight of an item is intimately related to the (sensitive) item itself. Thus their mechanisms are not adapted and applicable to this situation.

In another recent work, both Ebadi et al. (2015) and Proserpio et al. (2014) have proposed non-uniform extensions to PINQ (McSherry, 2009). More precisely, Ebadi et al. (2015) developed ProPer, an interactive system to track the privacy budget spent on each user's data. ProPer relies on the observation that in a sequence of PINQ queries, not all users' data are included in the query. Therefore, they avoid to spend the budget on users that are not included in a query. Heterogeneity is achieved when user records exceeding their privacy budget are silently dropped, while the other records that have not yet exhausted their budget are kept. This method is applicable only to a sequence of (live) queries, but not to a single query. In this situation, each record can have a different privacy budget, roughly related to its privacy weight, but this privacy weight will be public like (Jorgensen et al., 2015). In addition, one of the limits is that they assume that the result of queries are discrete and finite. Instead, Proserpio et al. (2014) reduce the total amount of noise needed for high-sensitivity transformations (such as PINQ Join query). In particular, they make such transformations "stable') by scaling down the value of records that strongly affect the result. The scaling weights are not selected by the user but rather by an algorithm aiming at keeping the sensitivity of all records "close" to each. In this work, the weights are not intended to reflect varying levels of privacy. Thus, it is unclear whether their work can be applied for

HDP, although it is reminiscent of the STRETCHING mechanism in the sense that it adjusts the sensitivity of the query.

6 Conclusion

In this work, we have introduced the novel concept of *heterogeneous differential privacy* that can accommodate for different privacy expectations not only per user, but also per item as opposed to previous models that implicitly assume uniform privacy requirements. We have also described a generic mechanism achieving HDP called the *Stretching Mechanism*, which protects at the same time the items of the profile of user and the privacy vector representing his privacy expectations across items of the profile. We applied this mechanism for the computation of the cosine similarity and evaluate its impact on a distributed semantic clustering task by using the recall as a measure of utility. Moreover, we have conducted an experimental evaluation of the impact of having different groups of users with different privacy requirements.

Although the Stretching Mechanism can be applied to a wealth of functions, it is nonetheless not directly applicable to some natural functions, such as the ℓ_0 norm and min. Indeed, when computing the ℓ_0 norm (*i.e.*, the number of non-zero coordinates in a given vector), each coordinate contributes either zero or one regardless of its value. Since the Stretching Mechanism modifies this value, this mechanism would always output the true exact value as long as no privacy weight has been set to exactly zero. For the case of min, due to the fact that the Stretching Mechanism shrinks each coordinate by a factor corresponding to its privacy weight, the resulting output may not have any relation to the intended semantics of the function min.

Another challenge is to enable users to estimate the amount of distortion in the output that they received out of an heterogenous differentially private mechanism. For instance, for functions such as the sum, recipients will not be able to estimate the correct value without being given the distortion. Although the distortion has an upper bound given by Lemma 3, the information needed to compute the upper bound is private. Therefore, releasing the distortion (or even its upper bound) would constitute a violation of privacy. We believe this issue could be solved partially by releasing an upper bound using the traditional Laplacian mechanism at an additional cost of an ε amount of privacy. Another important future work includes the characterization of functions that have a low and high distortion. Indeed, functions having a high distortion are not really suitable for our HDP mechanism. We also leave as open the question of designing a different mechanism than the Stretching Mechanism achieving HDP with a lower distortion.

References

Alaggan, M., Gambs, S., and Kermarrec, A.-M. (2011). "Private Similarity Computation in Distributed Systems: From Cryptography to Differential Privacy." In Anta, A. F., Lipari, G., and Roy, M. (eds.), *Proceedings of the 15th International Conference on the Principles of Distributed Systems (OPODIS'11)*, volume 7109 of *Lecture Notes in Computer Science*, 357–377. Toulouse, France: Springer.

— (2012). "BLIP: Non-Interactive Differentially-Private Similarity Computation on Bloom Filters." In *Proceedings of the 14th International Symposium on Stabilization, Safety, and Security of Distributed Systems (SSS'12)*. Toronto, Canada. To appear.

Beimel, A., Nissim, K., and Omri, E. (2011). "Distributed Private Data Analysis: on Simultaneously Solving *How* and *What*." *CoRR*, abs/1103.2626.

Bertier, M., Frey, D., Guerraoui, R., Kermarrec, A.-M., and Leroy, V. (2010). "The Gossple Anonymous Social Network." In *Proceedings of the 11th International Middleware Conference (Middleware'10), ACM/IFIP/USENIX*, 191–211. Bangalore, India.

Dandekar, P., Fawaz, N., and Ioannidis, S. (2011). "Privacy Auctions for Inner Product Disclosures." *CoRR*, abs/1111.2885.

Das, K., Bhaduri, K., and Kargupta, H. (2011). "Multi-Objective Optimization Based Privacy Preserving Distributed Data Mining in Peer-to-Peer Networks." *Peer-to-Peer Networking and Applications*, 4(2): 192–209.

Dwork, C. (2008). "Differential Privacy: a Survey of Results." In Agrawal, M., Du, D.-Z., Duan, Z., and Li, A. (eds.), *Proceedings of the 5th International Conference on Theory and Applications of Models of Computation (TAMC'08)*, 1–19. Xi'an, China.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). "Calibrating Noise to Sensitivity in Private Data Analysis." In *Proceedings of the 3rd Theory of Cryptography Conference (TCC'06)*, 265–284. New York, USA.

Dwork, C. and Naor, M. (2010). "On the Difficulties of Disclosure Prevention in Statistical Databases or the Case for Differential Privacy." *Journal of Privacy and Confidentiality*, 2(1): 93–107.

Dym, H. (2007). *Linear Algebra in Action*. Weizmann Institute of Science - AMS.

Ebadi, H., Sands, D., and Schneider, G. (2015). "Differential Privacy: Now it's Getting Personal." In Rajamani, S. K. and Walker, D. (eds.), *Proceedings of the 42nd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, (POPL'15)*, 69–81. Mumbai, India: ACM.
URL http://dl.acm.org/citation.cfm?id=2676726

Ghosh, A. and Roth, A. (2011). "Selling Privacy at Auction." In Shoham, Y., Chen, Y., and Roughgarden, T. (eds.), *Proceedings of the 12th ACM Conference on Electronic Commerce (EC-2011)*, 199–208. San Jose, CA, USA: ACM.

Harris Interactive (2003). "The Harris Poll® ♯17: Most People are 'Privacy Pragmatists' who, while Concerned about Privacy, will Sometimes Trade it off for Other Benefits." URL http://www.harrisinteractive.com/harris_poll/index.asp?PID=365

Jeffrey, A. (2010). *Matrix Operations for Engineers and Scientists: An Essential Guide in Linear Algebra*, chapter Linear Transformations and the Geometry of the Plane, 239–272. Springer Netherlands.

Jensen, C., Potts, C., and Jensen, C. (2005). "Privacy Practices of Internet Users: Self-Reports versus Observed Behavior." *International Journal of Human-Computer Studies*, 63(1-2): 203–227.

Jorgensen, Z., Yu, T., and Cormode, G. (2015). "Conservative or Liberal? Personalized Differential Privacy." In *31st IEEE International Conference on Data Engineering (ICDE'15)*.

Kasiviswanathan, S. P., Lee, H. K., Nissim, K., Raskhodnikova, S., and Smith, A. (2008). "What Can We Learn Privately?" In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS'08)*, 531–540. IEEE Computer Society Washington, DC, USA.

Kumar, R., Gopal, R. D., and Garfinkel, R. S. (2010). "Freedom of Privacy: Anonymous Data Collection with Respondent-Defined Privacy Protection." *INFORMS Journal on Computing*, 22(3): 471–481.

Lee, J. and Clifton, C. (2011). "How Much is Enough? Choosing ϵ for Differential Privacy." In Lai, X., Zhou, J., and Li, H. (eds.), *Proceedings of the 14th International Information Security Conference (ISC'11)*, volume 7001 of *Lecture Notes in Computer Science*, 325–340. Xi'an, China: Springer.

Liu, F., Yu, C., and Meng, W. (2004). "Personalized Web Search for Improving Retrieval Effectiveness." *IEEE Transactions on Knowledge and Data Engineering*, 16(1): 28 – 40.

Liu, Y., Gummadi, K. P., Krishnamurthy, B., and Mislove, A. (2011). "Analyzing Facebook Privacy Settings: User Expectations vs. Reality." In *Proceedings of the Internet Measurement Conference*, 61–70. Berlin, Germany.

McGregor, A., Mironov, I., Pitassi, T., Reingold, O., Talwar, K., and Vadhan, S. (2011). "The Limits of Two-Party Differential Privacy." Technical Report 106, Electronic Colloquium on Computational Complexity.

McSherry, F. and Mironov, I. (2009). "Differentially Private Recommender Systems: Building Privacy into the Net." In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09)*, 627–636. New York, NY, USA: ACM.

McSherry, F. D. (2009). "Privacy Integrated Queries: an Extensible Platform for Privacy-Preserving Data Analysis." In *Proceedings of the 2009 ACM International Conference on Management of Data (SIGMOD'09)*, 19–30. ACM.

Mironov, I., Pandey, O., Reingold, O., and Vadhan, S. P. (2009). "Computational Differential Privacy." In Halevi, S. (ed.), *Proceedings of the 29th Annual International Cryptography Conference – Advances in Cryptology (CRYPTO'09)*, volume 5677 of *Lecture Notes in Computer Science*, 126–142. Santa Barbara, CA, USA: Springer.

Nissim, K., Raskhodnikova, S., and Smith, A. (2007). "Smooth Sensitivity and Sampling in Private Data Analysis." In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing (STOC'07)*, 75–84. San Diego, California, USA.

Preibusch, S. and Beresford, A. R. (2009). "Privacy-Preserving Friendship Relations for Mobile Social Networking." In *Proceedings of the W3C Workshop on the Future of Social Networking*. Barcelona, Spain.

Proserpio, D., Goldberg, S., and McSherry, F. (2014). "Calibrating Data to Sensitivity in Private Data Analysis." *40th International Conference on Very Large Data Bases (PVLDB'14)*, 7(8): 637–648.

Sweeney, L. (2002). "k-Anonymity: A Model for Protecting Privacy." *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5): 557–570.

Toch, E., Wang, Y., and Cranor, L. (2012). "Personalization and Privacy: a Survey of Privacy Risks and Remedies in Personalization-Based Systems." *User Modeling and User-Adapted Interaction*, 22(1-2): 203–220. 10.1007/s11257-011-9110-z.

Venkatasubramanian, S. (2008). *Privacy-Preserving Data Mining*, volume 34 of *Advances in Database Systems*, chapter Measures of Anonymity, 81–103. Springer US.

Wen, Z. and Lin, C.-Y. (2010). "How Accurately Can One's Interests Be Inferred from Friends?" In *Proceedings of the 19th International Conference on World Wide Web*, WWW'10, 1203–1204. New York, NY, USA: ACM.

Zeng, Y., Zhong, N., Ren, X., and Wang, Y. (2012). "User Interests Driven Web Personalization Based on Multiple Social Networks." In *Proceedings of the 4th International Workshop on Web Intelligence & Communities*, WI&C'12, 9:1–9:4. New York, NY, USA: ACM.

Zhang, N. and Zhao, W. (2007). "Privacy-Preserving Data Mining Systems." *Computer*, 40(4): 52–58.

Zhou, X., Xu, Y., Li, Y., Josang, A., and Cox, C. (2012). "The State-of-the-Art in Personalized Recommender Systems for Social Networking." *Artificial Intelligence Review*, 37(2): 119–132.

Zwick, D. and Dholakia, N. (1999). "Models of Privacy in the Digital Age: Implications for Marketing and E-Commerce."

1 Proof of Lemma 1

Proposition 2 (Monotonicity of subdomain optimization)**.** *Let θ and θ' be the result of two maximization problems p_1 and p_2 of the function g in which the maximization is over domains J and J', respectively. Then, if $J \subseteq J'$, this implies that $\theta \leqslant \theta'$. The opposite statement also holds for minimization problems.*

Proof 11. *Since θ' is the optimal result of p_2 over J', this means that by definition:*

$$g(\theta') \geqslant g(j) \text{ , for all } j \text{ in } J'. \tag{17}$$

Moreover, since any result θ for p_1 will always be in J, and therefore in J', then $g(\theta') \geqslant g(\theta)$ by (17). (The proof that the opposite statement holds for minimization problems follows from the same arguments and thus we choose to omit it.) □

Lemma 6 (Shrinkage matrices composition)**.** *If A and B are two shrinkage matrices and D a semi-balanced set, then $ABD \subseteq BD \subseteq D$.*

Proof 12. *By definition of semi-balanced set, we have $BD \subseteq D$. Then it remains to prove that $ABD \subseteq BD$ (or equivalently, that BD is a semi-balanced set). We observe that a vector $\vec{w}$ belongs to ABD if and only if $\vec{w} = AB\vec{b}$ for some $\vec{b} \in D$. Because shrinking matrices commute, $\vec{w} = BA\vec{b}$. Let $\vec{a} = A\vec{b}$. By definition of semi-balanced set, $\vec{a} \in D$. Therefore, $\vec{w} = B\vec{a}$ for $\vec{a} \in D$, which means $\vec{w}$ belongs to BD by definition of BD.* □

Lemma 7 (Monotonicity of the global sensitivity)**.** *If $\vec{w}' \leqslant \vec{w}$ then $S(R, \vec{w}') \leqslant S(R, \vec{w})$.*

Proof 13. *Let $\vec{c}$ be such that*

$$c_i = \begin{cases} w'_i / w_i & \text{if } w_i \neq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{18}$$

and let $C = \operatorname{diag}(\vec{c})$ is a shrinkage matrix. Then $\vec{w}' = C\vec{w}$. Let $T' = \operatorname{diag}(\vec{w}')$ and $T = \operatorname{diag}(\vec{w})$ be two other shrinkage matrices. Notice that $T' = CT$. By Lemma 6 and since D is semi-balanced:

$$T'D = CTD \subseteq TD \subseteq D. \tag{19}$$

The result follows from Proposition 2 because $S(R, \vec{w})$ is over the domain TD while $S(R, \vec{w}')$ is a maximization problem over the domain $T'D \subseteq TD$. □

Corollary 3 (Monotonicity of the modular global sensitivity)**.** *If $\vec{w}' \leqslant \vec{w}$ then $S_i(R, \vec{w}') \leqslant S_i(R, \vec{w})$ for all i.*

Proof 14. *By Lemma 7, we have that $S(R, \vec{w}') \leqslant S(R, \vec{w})$. Let $i^* = \operatorname{argmax}_i S_i(R, \vec{w})$ and therefore $S(R, \vec{w}) = S_{i^*}(R, \vec{w})$. In order to get a contradiction, we assume that $S_{i^*}(R, \vec{w}') > S_{i^*}(R, \vec{w})$, thus we have*

$$S(R, \vec{w}') = \max_i S_i(R, \vec{w}') > S_{i^*}(R, \vec{w}) = S(R, \vec{w}), \tag{20}$$

which is a contradiction. □

Proof 15 (Proof of Lemma 1)**.** *Since* $\vec{w} \leqslant (\vec{1}_{-i}, w_i)$ *for all* i*, then:*

$$\begin{aligned} S_i(R, \vec{w}) &\leqslant S_i(R, (\vec{1}_{-i}, w_i)) \text{ for all } i, \\ &\leqslant v_i S(f) \quad \text{for all } i, \end{aligned}$$

where the first inequality follows by Corollary 3 and the second inequality follows from the premise of the lemma, thus concluding the proof. □