Lightweight Privacy-Preserving Task Assignment in Skill-Aware Crowdsourcing

Louis Béziaud, Tristan Allard, David Gross-Amblard

Lightweight Privacy-Preserving Task Assignment in Skill-Aware Crowdsourcing [Full Version]

Louis Béziaud[1,3], Tristan Allard[1,2], and David Gross-Amblard[1,2]

[1] Univ. Rennes 1, France
[2] IRISA, France, first.last@irisa.fr
[3] ENS Rennes, France, first.last@ens-rennes.fr

Abstract. Crowdsourcing platforms dedicated to work, be it paid or voluntary, essentially consist in intermediating between tasks—sent by requesters—and workers. They are used by a growing number of individuals and organizations, for tasks that are more and more diverse, complex, and that require specific skills, availabilities, experiences, or even devices. On the one hand, these highly detailed task specifications and worker profiles enable high-quality task assignments. On the other hand, detailed worker profiles may disclose a large amount of personal information to the central platform (*e.g.,* personal preferences, availabilities, wealth, occupations), jeopardizing the privacy of workers. In this paper, we propose a lightweight approach to protect workers privacy against the platform along the current crowdsourcing task assignment process. Our approach (1) satisfies differential privacy by building on the well-known randomized response technique, applied by each worker to perturb locally her profile before sending it to the platform, and (2) copes with the resulting perturbation by benefiting from a taxonomy defined on workers profiles. We describe the lightweight upgrades to be brought to the workers, to the platform, and to the requesters. We show formally that our approach satisfies differential privacy, and empirically, through experiments performed on various synthetic datasets, that it is a promising research track for coping with realistic cost and quality requirements.

Keywords: Crowdsourcing; Task assignment; Differential privacy; Randomized response

1 Introduction

Crowdsourcing platforms are disrupting traditional work marketplaces. Their ability to compute high-quality matchings between tasks and workers, instantly and worldwide, for paid or voluntary work, has made them unavoidable actors of the 21^{st} century economy. Early crowdsourcing platforms did not (and still do not) require strong and specific skills; they include for example Amazon Mechan-

ical Turk[4] (for online micro-tasks), Uber[5] (for car-driving tasks), or TaskRabbit[6] (for simple home-related tasks—*e.g.,* cleaning, repairing). Today's crowdsourcing platforms now go one step further by addressing skill-intensive contexts (*e.g.,* general team building[7], collaborative engineering[8]) through the collection and use of fine-grained worker profiles. Such platforms carry the promise to facilitate, fasten, and spread innovation at an unprecedented scale.

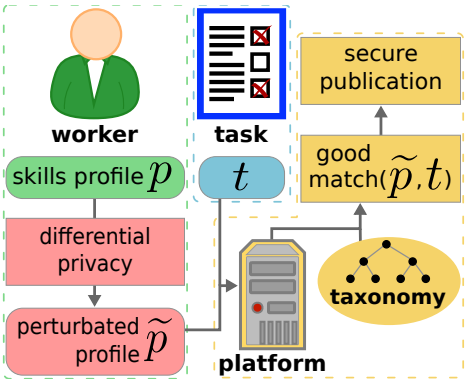

Fig. 1: Our Approach to Privacy-Preserving Task Assignment

However abusive behaviors from crowdsourcing platforms against workers are frequently reported in the news or on dedicated websites, whether performed willingly or not (see, *e.g.,* the privacy scandals due to illegitimate accesses to the geolocation data of a well-known drivers-riders company[9], or the large-scale exposure of workers' identifying and sensitive information—*e.g.,* real name, book reviews, or wish-list—through Amazon Mechanical Turk IDs [12]). The problem is even more pregnant with skill-intensive crowdsourcing platforms since they collect detailed workers' profiles for computing highly accurate matchings (*e.g.,* demographics, encompassive set of skills, detailed past experiences, personal preferences, daily availabilities, tools possessed). Even without considering fully the wealth of information in a worker's profile, a detailed set of skills together with the worker's location can already disclose significantly identifying and/or sensitive information (*e.g.,* the only `expert in fully homomorphic encryption` with `advanced knowledge in hatha yoga` from the `city of Rennes` also has a strong background in `alternative medicines for alleviating the side-effects of chemotherapy` may reflect a health status—of the worker or of a relative—and be identifying). In some cases, even though a worker's profile does not contain any skill considered to be "sensitive", this is

[4] https://www.mturk.com/
[5] https://www.uber.com/
[6] https://www.taskrabbit.com/
[7] https://tara.ai/
[8] https://makake.co/
[9] https://tinyurl.com/wp-priv

the act of participating to crowdsourced projects that is sensitive (*e.g.,* a company may prefer that its employees *do not participate* to crowdsourced projects in order to avoid any risk of disclosures). We advocate thus for a sound protection of workers' profiles against illegitimate uses: in addition to the necessary compliance with fundamental rights to privacy, it is a precondition for a wide adoption of crowdsourcing platforms by individuals.

Computing the assignment of tasks to workers is the foundamental role of the platform (or at least facilitating it). This paper considers precisely the problem of computing a high-quality matching between skill-intensive tasks and workers while preserving workers' privacy. To the best of our knowledge, this problem has only been addressed by a single recent work [8] (see Section 5 for other related works). However, this work is based on costly homomorphic encryption primitives which strongly hamper its performances and prevent it to reason about skills within the assignment algorithm (*e.g.,* no use of semantic proximity).

In this paper, we propose an approach depicted in Fig. 1 that addresses these issues by making the following contributions:

1. A simple skills model for a worker's profile: a bit vector and a taxonomy.
2. An algorithm run independently by each worker for perturbing her profile locally before sending it to the platform. By building on the proven *randomized response* mechanism [3, 18], this algorithm is privacy-preserving (provides sound *differential privacy* guarantees [7, 19]), and lightweight (no cryptography, no distributed computation, only bitwise operations).
3. A suite of weight functions to be plugged in a traditional assignment algorithm run by the platform and dedicated to increase the quality of matchings performed over perturbed profiles. Our weight functions reduce the impact of the perturbation by leveraging the skills taxonomy, vertically and horizontally, averaging the skills according to their semantic proximity in order to reduce the variance of the differentially-private perturbation. The variance reduction is mathematically sound and does not jeopardize privacy.
4. An experimental study, over a synthetic taxonomy and various synthetic datasets, that shows promising preliminary results about the practical adequacy of our approach from the sides of performance and quality.

The rest of the paper is organized as follows. Section 2 introduces the notions used in our approach and defines more precisely the problem we tackle. We describe our algorithms in Section 3 and analyze them empirically in Section 4. Section 5 discusses related work. Finally, we conclude in Section 6, outlining interesting future works.

2 Problem Definition

2.1 Skills and Participants

We model below the three types of participants that interact together during a crowdsourcing process as well as the information (*i.e.,* skills) based on which the interaction takes place (see Fig. 1).

Skills Model The set of skills that can be possessed by a worker (*resp.* requested by a task) is denoted $\mathcal{S}$. A worker's profile $p_i \in \mathcal{P}$ (*resp.* a task $t_i \in \mathcal{T}$) is represented by a bit vector, *i.e.,* $p_i = \{0,1\}^{|\mathcal{S}|}$, where each bit corresponds to a skill $s_j \in \mathcal{S}$ and is set to 1 if the given worker has the given skill (*resp.* the given task $t_i = \{0,1\}^{|\mathcal{S}|}$ requests the given skill). Furthermore, we assume that a skills taxonomy $\mathcal{S}_T$ exists[10] [13], structuring the skills according to their semantic proximity, and is such that the skills in $\mathcal{S}$ are the leaves of $\mathcal{S}_T$ (*i.e.,* no non-leaf node can be possessed nor requested). The non-leaf nodes of the taxonomy are called *super-skills*. For simplicity, we omit here other kind of information (*e.g.,* personal preferences, location, age) but we emphasize that our techniques can be extended easily to support a wide range of information.

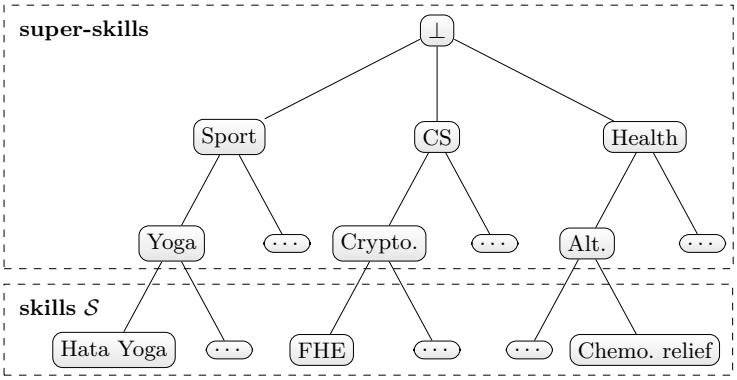

Fig. 2: Example of a skill taxonomy $\mathcal{S}_T$

Workers and Requesters Both workers and requesters have storage, computing, and communication ressources (*e.g.,* typical CPU/RAM/bandwidth of today's personal computers); they differ from their privacy constraints. Indeed, each worker has a private profile, while each requester has non-confidential tasks. Without loss of generality, we consider that each requester has a single task and that the number of workers and requesters is the same (*i.e.,* $|\mathcal{P}| = |\mathcal{T}|$).

Platform The platform is essentially in charge of intermediating between workers and requesters. It holds the set of workers' profiles, *perturbed* to satisfy differential privacy (defined below) and denoted $\widetilde{\mathcal{P}}$, as well as the exact set of requesters' tasks $\mathcal{T}$.

Attack Model All participants, *i.e.,* workers, requesters, and the platform, are considered to be *honest-but-curious*. This means that they participate in the protocol without deviating from its execution sequence (*e.g.,* no message tampering, no data forging) but they will try to infer anything that is computationally-feasible to infer about private data (*i.e.,* the set of workers' non-perturbed profiles $\mathcal{P}$).

[10] In practice, skills taxonomies concerning numerous real-life contexts exist today (see, *e.g.,* the Skill-Project http://en.skill-project.org/skills/, or Wand's taxonomies http://www.wandinc.com/wand-skills-taxonomy.aspx).

2.2 The Traditional Tasks-to-Workers Assignment Problem

In a traditional context, where workers' profiles are not considered private, the objective of the crowdsourcing platform is to assign a worker to each task such that the overall expected quality is maximized. This well-known combinatorial optimization problem is referred as the assignment problem and can be expressed as a standard linear problem [11] (assuming $|\mathcal{T}| = |\mathcal{P}|$). We define it formally as follows:

Problem 1 (Assignment Problem). Given two sets, $\mathcal{T}$ and $\mathcal{P}$, of equal size, and a weight function $\mathtt{C}\colon \mathcal{T} \times \mathcal{P} \to \mathbb{R}$, the assignment problem is to find a bijection $\mathtt{f}\colon \mathcal{P} \to \mathcal{T}$ such that the cost function $\sum_{t \in \mathcal{T}} \mathtt{C}(t, \mathtt{f}(t))$ is minimized.

Assignment algorithms rely on the *weight function* $\mathtt{C}\colon \mathcal{T} \times \mathcal{P} \to \mathbb{R}$ in charge of defining the cost of each assignment, *i.e.,* the divergence between the requirements vector of a task and the skills vector of a worker. Common weight functions include the usual distance metrics (*e.g.,* Hamming distance) or disimilarities (*e.g.,* cosine distance). Numerous algorithms exist for solving the assignment problem [1, 11, 13]. Since our approach is independent from the algorithm, we simply use the HUNGARIAN method [11], a standard academic choice. The HUNGARIAN method computes an optimal matching in $\mathcal{O}(n^3)$ running time. In a nutshell, it sets up all of the costs in a matrix, and uses a combination of row operations and zero covers to come up with a min-cost assignment, resulting in an optimal solution. We refer the interested reader to the original paper [11] for further details.

2.3 Security and Quality

Security We say that our approach is secure against honest-but-curious participants if and only if no participant learns information about the set of non-perturbed profiles $\mathcal{P}$ that has not been perturbed by a differentially-private mechanism, where differential privacy is defined below.

Differential privacy [5] is the current *de facto* standard model for disclosing personal information while satisfying sound privacy guarantees. Informally speaking, the differential privacy model requires that the outputs of a differentially-private algorithm be almost insensitive to the participation of any possible individual in the dataset input. Differential privacy is composable and secure under post-processing.

Definition 1 (Differential Privacy [5]). *A randomized mechanism* $\mathtt{M}$ *satisfies* ϵ*-differential privacy with* $\epsilon > 0$ *if for any possible set of workers' profiles* $\mathcal{P}$ *and* $\mathcal{P}'$ *such that* $\mathcal{P}'$ *is* $\mathcal{P}$ *with one additional profile (or one profile less), and any possible set of output* $O \subseteq \mathit{Range}(\mathtt{M})$*,*

$$\Pr[\mathtt{M}(\mathcal{P}) \in O] \leq e^{\epsilon} \times \Pr[\mathtt{M}(\mathcal{P}') \in O]. \tag{1}$$

Theorem 1 (Compositions [14]). *Let* $\mathtt{M}$ *be a randomized mechanism which satisfies* ϵ*-differential privacy. Then: (1) executing* $\mathtt{M}$ *i times sequentially over*

the dataset $\mathcal{D}$, each time with a privacy parameter set to ϵ_i, provides $(\sum_i \epsilon_i)$-differential privacy (sequential composition), and (2) executing $\mathtt{M}$ *over i disjoint subsets of the dataset $\mathcal{D}$, each time with a privacy parameter set to ϵ_i, provides $(\max_i \epsilon_i)$-differential privacy (parallel composition).*

Theorem 2 (Post-Processing [7]). *Let* $\mathtt{M}$ *be a randomized mechanism which provides ϵ-differential privacy. Let* $\mathtt{f}$ *be an arbitrary randomized mapping. Then* $\mathtt{f} \circ \mathtt{M}$ *provides ϵ-differential privacy.*

Quality The inherent information loss due to the differentially-private perturbation impacts the quality of the worker-to-task assignment. This is the price to pay to satisfy a sound privacy model. We quantify this impact by measuring the relative increase of the assignment cost. The *relative quality* is the inverse of this increase:

Definition 2 (Relative Quality). *Let $\widetilde{\mathcal{A}}$ (resp. $\mathcal{A}$) be the assignment performed over the set of perturbed profiles $\widetilde{\mathcal{P}}$ (resp. the set of non-perturbed profiles $\mathcal{P}$). We define the relative quality as follows:*

$$\mathtt{q}_{\mathtt{rel}} = \frac{\sum_{(t,p)\in\mathcal{A}} \mathtt{C}(t,p)}{\sum_{(t,\widetilde{p})\in\widetilde{\mathcal{A}}} \mathtt{C}(t,\widetilde{p})} \tag{2}$$

We complement the relative quality with the fraction of profiles that have *all* the skills required by the task to which they are assigned:

Definition 3 (Fraction of Perfect Assignments). *Let $\widetilde{\mathcal{A}}$ be the assignment performed over the set of perturbed profiles $\widetilde{\mathcal{P}}$. We define the fraction of perturbed assignments as follows:*

$$\mathtt{fpa} = 1/|\widetilde{A}| \times \sum_{\forall i} (t[i] = 1 \wedge p[i] = 1) \vee t[i] = 0 \tag{3}$$

3 Flip-based Approach

This section details our approach. We first focus on the workers' side: we describe the algorithm that we propose for perturbing each worker's profile, show its adequacy to our context, and demonstrate that it complies with our security model. Second, we shift to the platform's side. We explain how to reduce the impact of the differentially private perturbation (while still satisfying differential privacy) and we describe the assignment algorithm based on perturbed profiles. Finally, we overview technical means for letting workers fetch their assignment in a secure way in order to complete it.

3.1 At a Worker's Side: Local Perturbation

Building Block: Randomized Response Randomized response [18] is a simple though powerful perturbation mechanism shown to satisfy differential privacy

(see below). Basically, it inputs a single bit (*e.g.,* the answer of an individual to a sensitive boolean question) and flips it randomly according to a well-chosen distribution probability. We advocate its use in our approach for perturbing workers' profiles because it can be performed by each worker independently from the others without any need for cryptography nor network communication (beyond the necessary transfer of each perturbed profile to the platform). This would not be the case with other well-known differentially-private mechanisms, such as the addition of Laplace noise [5] for example, which require to aggregate data before perturbing it (usually by centralizing it).

There exist multiple variants of the randomized response mechanism, some of them in the context of differential privacy [7, 19]. We describe below the variant, called *innocuous question*, we use in this paper and show that it satisfies differential privacy[11]. Let $x \in \{0, 1\}$ be a private value. The randomized response mechanism simply outputs the perturbed value of x, denoted $\widetilde{x}$, as follows:

$$\widetilde{x} = \begin{cases} x & \text{with probability } 1 - \Pr_{flip} \\ 1 & \text{with probability } \Pr_{flip} \times \Pr_{inno} \\ 0 & \text{with probability } \Pr_{flip} \times (1 - \Pr_{inno}) \end{cases}$$

where Pr_{flip} depends on ϵ (see below) and $Pr_{inno} \in [0, 1]$. We use $\Pr_{inno} = 0.5$ in the rest of the paper, since it minimizes the variation of the estimated value of x after perturbation (see [3] for more details).

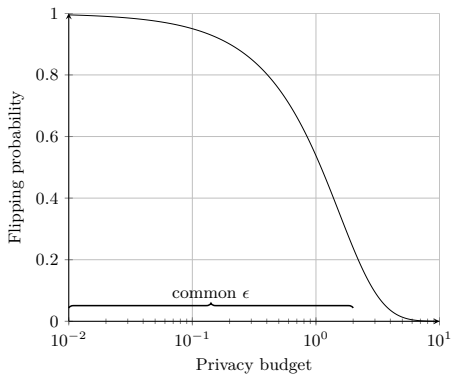

Fig. 3: Privacy budget ϵ with respect to the flipping probability $\Pr_{flip}$.

Claim. For a given differential privacy parameter $\epsilon > 0$, and a worker's profile made of a single bit to be flipped, the innocuous question randomized response scheme satisfies ϵ-differential privacy if $\Pr_{flip} = \frac{2}{1+e^{\epsilon}}$.

Proof. Let $\Pr[\widetilde{x} = u \mid x = v]$ $(u, v \in \{0, 1\})$ denote the probability that the random output $\widetilde{x}$ is u when the real attribute value x is v. The innocuous question randomized response scheme gives the following probabilities:

[11] Any other variant could have been used, provided that it satisfies differential privacy.

Algorithm 1: FLIP (run by each Worker)

Input: The original profile $p = \langle p[1], \ldots, p[l] \rangle$, the differential privacy budget $\epsilon > 0$.

Let $\texttt{Pr}_{flip}$ be the flipping probability: $\Pr_{flip} \leftarrow \frac{2}{1+e^{\epsilon/l}}$.
Initiate the perturbed profile: $\widetilde{p} \leftarrow \langle \widetilde{p}[1] = 0, \ldots, \widetilde{p}[l] = 0 \rangle$.
for $1 \leq i \leq l$ **do**
  $\widetilde{p}[i] \leftarrow \texttt{RandomizedResponse}(p[i], \Pr_{flip})$.
Return *The perturbed profile* $\widetilde{p}$

(1) $\Pr[\widetilde{x} = 0 \mid x = 0] = (1 - \Pr_{flip}) + \Pr_{flip} * (1 - \Pr_{inno})$
(2) $\Pr[\widetilde{x} = 0 \mid x = 1] = \Pr_{flip} * (1 - \Pr_{inno})$
(3) $\Pr[\widetilde{x} = 1 \mid x = 0] = \Pr_{flip} * \Pr_{inno}$
(4) $\Pr[\widetilde{x} = 1 \mid x = 1] = (1 - \Pr_{flip}) + (\Pr_{flip} * \Pr_{inno})$

In order to satisfy ϵ-differential privacy, we must ensure that $\frac{\Pr[\widetilde{x}=u|x=u]}{\Pr[\widetilde{x}=u|x=v]} \leq e^{\epsilon}$. This translates into $\frac{\Pr[\widetilde{x}=0|x=0]}{\Pr[\widetilde{x}=0|x=1]} \leq e^{\epsilon}$ and $\frac{\Pr[\widetilde{x}=1|x=1]}{\Pr[\widetilde{x}=1|x=0]} \leq e^{\epsilon}$. Given the randomized response probabilities above and the value of $\Pr_{inno} = 0.5$, the two unequalities are satisfied (and maximized) for $\Pr_{flip} = \frac{2}{1+e^{\epsilon}}$.

Flip Mechanism Our FLIP mechanism essentially consists in applying the randomized response mechanism to each binary skill of a worker's profile before sending it to the platform and inherits thus its high efficiency. Due to Theorem 1, the privacy parameter ϵ must be distributed over the bits of the skills vector (*i.e.,* divided by the number of bits) to compute the bit-flipping probability[12]. We detail the FLIP mechanism in Algorithm 1 and show that it satisfies ϵ-differential privacy.

Claim. The FLIP mechanism satisfies ϵ-differential privacy.

Proof. The proof follows the composition properties of differential privacy. First, let consider that we have a single worker with a skills vector p of length l. We showed above that perturbing (and sending to the platform) a single bit of p with a flipping probability $\Pr_{flip} = 2/(1 + e^{\epsilon/l})$ satisfies $\frac{\epsilon}{l}$-differential privacy. It results from Theorem 1 that perturbing l bits of p all with probability $\Pr_{flip} = 2/(1 + e^{\epsilon/l})$ yields $(\sum_i^l \frac{\epsilon}{l})$-differential privacy, *i.e.,* ϵ-differential privacy. Hence, with a single worker, *i.e.,* $|\widetilde{\mathcal{P}}| = 1$, the FLIP mechanism satisfies ϵ-differential privacy. Let now consider an arbitrary number of workers, say n. Since each perturbed profile $p_i \in \widetilde{\mathcal{P}}$ consists in a disjoint subset of $\widetilde{\mathcal{P}}$, then it follows directly from Theorem 1 that the FLIP mechanism applied in parallel by each worker

[12] The FLIP mechanism could limit the distribution of ϵ by sampling the skills vector uniformly at random before perturbing it. The benefitial impact of sampling on differential privacy has already been studied in other contexts [10]. We let the study of this extension to future works.

satisfies $\max_i \epsilon_i$-differential privacy. Moreover, all workers use the same privacy budget $\epsilon_i = \epsilon$, so that $\max_i \epsilon_i = \epsilon$. As a result, the parallel execution of the FLIP mechanism by the n workers (*i.e.,* $|\widetilde{\mathcal{P}}| = n$) satisfy ϵ-differential privacy.

3.2 At the Platform's Side: Task Assignment

Algorithm 2: MATCH (run by the Platform)

Input: The set of perturbed profiles $\widetilde{\mathcal{P}}$ (bit vectors), the set of tasks $\mathcal{T}$ (bit vectors), the weight function $\mathtt{C}\colon \mathcal{T} \times \mathcal{P} \to \mathbb{R}$ to be used.

1 Initiate the final assignments $\widetilde{\mathcal{A}}$.
2 Perform the HUNGARIAN method, based on $\mathtt{C}$, for computing the task-to-worker assignment: $\widetilde{\mathcal{A}} \leftarrow \mathtt{Hungarian}(\widetilde{\mathcal{P}}, \mathcal{T}, \mathtt{C})$.
3 **Return** *The assignment* $\widetilde{\mathcal{A}}$

Efficient traditional assignment algorithms do not need any modification to work with perturbed profiles, which are bit vectors, exactly as non-perturbed profiles are. The MATCH algorithm presented in Algorithm 2 shows our straightforward use of the HUNGARIAN assignment algorithm. The main question is the impact on quality due to our perturbation, and this impact is naturally related to the weight function $\mathtt{C}\colon \mathcal{T} \times \mathcal{P} \to \mathbb{R}$ on which assignment algorithms rely. As there is no clear consensus on what is a good weight function for task assignment, in the sequel we recall several reasonable functions, ignoring or using the skill taxonomy. We also propose new weight functions and explain how they could cope with the diffentially-private perturbation.

Existing Weight Functions Numerous weight functions have been proposed as metrics over skills. The *Hamming distance* is a common choice to compute dissimilarity between two vectors of bits. The *Hamming distance* between two vectors is the number of positions at which the corresponding symbols are different. However it does not capture the semantics of requirement needed for crowdsourcing: *e.g.*, a worker possessing every skills has a high *Hamming distance* from a task requiring only one skill, in spite of the fact that he is perfectly capable of achieving this task. The weight function proposed in [13] adresses this problem based on a taxonomy. We slightly adapt it and call it the `Ancestors` weight function (`AWF` for short).

Definition 4 (`Ancestors` Weight Function (adapted from [13])). *Let d_{max} be the maximum depth of the taxonomy $\mathcal{S}^T$. Let $\mathrm{lca}(s, s') \in \mathcal{S}^T$ be the lower common ancestor of skills s and s' in the taxonomy.*

$$\mathtt{AWF}(\widetilde{p}, t) = \sum_{s_i \in \widetilde{p}} \min_{s_j \in t} \left(\frac{d_{max} - \mathrm{depth}(\mathrm{lca}(s_i, s_j))}{d_{max}} \right) \tag{4}$$

Naive Skill-Level Weight Functions The `Missing` weight function (`MWF` for short) between a worker and a task revisits the Hamming distance. It settles a task-to-worker assignment cost that is intuitive in a crowdsourcing context: it is defined as the fraction of skills required by the task that the worker does not have (see Definition 5).

Definition 5 (`Missing` Weight Function (`MWF`)). $\texttt{MWF}\colon \mathcal{T} \times \widetilde{\mathcal{P}} \to \mathbb{R}$ *is defined as follows:*

$$\texttt{MWF}(t, \widetilde{p}) = \sum_{\forall i} t[i] \wedge \neg \widetilde{p}[i] \tag{5}$$

Leveraging the Taxonomy In realistic contexts, the differentially private perturbation may overwhelm the information contained in the original profiles and make the perturbed profiles be close to uniformly random bit vectors (*i.e.,* $\texttt{Pr}_{flip}$ close to 0.5). With a naive weight function, MATCH would assign tasks to random profiles, which would naturally result in random assignments. We cope with this challenging issue by building on the taxonomy $\mathcal{S}_{\texttt{T}}$. Indeed, the taxonomy allows to group large numbers of skills according to their semantic proximity and to reduce the perturbation magnitude by using group averages. Such a reduction is mathematically sound and formally assessed by the Bienaymé formula [2]. Informally speaking, Bienaymé states that the variance of the mean of a set of uncorrelated random variables decreases linearly with the number of variables averaged. In our context this means that the more the number of perturbed skills averaged together, the less the magnitude of the differentially private perturbation (linear decrease).

We propose below two weight functions designed to cope with the perturbation introduced by differential privacy by benefiting from Bienaymé formula. Each leverages a precise type of information given by the taxonomy: the *vertical parent-to-children relationship* between skills, and the *horizontal neighbouring proximity relationship*. We stress that our two weight functions aim at illustrating simply the benefits of using these two relationships. More complex alternative functions benefiting differently from the same information could be designed.

Climbing Weight Function The `Climbing` weight function (`CWF` for short) leverages *the vertical relationship* given by the taxonomy by averaging, for each profile, the skills along the root-to-leaf paths. In other words, before performing the assignment, the platform converts each perturbed profile into a tree[13], *i.e.,* the same as the taxonomy S_T, and for each node n of the tree, it computes the mean of the skills that appear below n (interpreting the boolean values 1 and 0 as integers). For a given node, this mean is actually a rough estimator of the fraction of descendant skills possessed. We call it *score* below. During an assignment, given a task and a perturbed profile, the `Climbing` weight function consists in computing the distance between the scores vector of the task and the

[13] Converting the perturbed profiles in trees of scores can be performed in parallel by each worker and be sent to the platform together with her flipped profile.

scores vector of the profile at each level, in weighting it by the level (the closer to the leaves the larger the weight), and finally in summing the whole. The variance reduction of the perturbation that `Climbing` is expected to benefit from is due to the average computed within each score. For each given score, the reduction is linearly proportional with the number of skills that it averages, which depends on the taxonomy used, but grows monotonically and inversely linearly with the level. Definition 6 formalizes the `Climbing` weight function.

Definition 6 (`Climbing` Weight Function (`CWF`)). *Let v_i (*resp. *u_i) denote the scores vector at level i in the tree corresponding to the profile $\widetilde{p}$ (*resp. *to the task t), and $\mathtt{d}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a classical distance function on real-valued vectors (*e.g., *`Cosine`). Then, $\mathtt{CWF}\colon \mathcal{T} \times \widetilde{\mathcal{P}} \to \mathbb{R}$ is defined as follows:*

$$\mathtt{CWF}(t, \widetilde{p}) = \sum_{\forall i} i \times \mathtt{d}(u_i, v_i) \tag{6}$$

Touring Weight Function The `Touring` weight function (`TWF` for short) leverages *the horizontal relationship* given by the taxonomy, *i.e.,* the neighbouring proximity degree between skills. As described in Definition 7, it returns the average path-length— according to the taxonomy $\mathcal{S}_T$—between the skills required by the task and the skills of the worker's profile. The expected variance reduction comes from the average path-length of the full cartesian product between the skills required by a task and the skills set to 1 in a perturbed profile. The reduction depends on the taxonomy (similarly to `Climbing`) and on the number of skills averaged.

Definition 7 (`Touring` Weight Function (`TWF`)). *Let $\rightsquigarrow$ denote the path-length operator between two skills in the taxonomy $\mathcal{S}_T$. Then, $\mathtt{TWF}\colon \mathcal{T} \times \widetilde{\mathcal{P}} \to \mathbb{R}$ is defined as follows:*

$$\mathtt{TWF}(t, \widetilde{p}) = \frac{\sum_{\forall i} \sum_{\forall j} (t[i] \wedge \widetilde{p}[j]) \times (s_i \rightsquigarrow s_j)}{\sum_{\forall i} \widetilde{p}[i] \times \sum_{\forall i} t[i]} \tag{7}$$

3.3 Post-Assignment Phase

Workers need a secure way to fetch their own assignment. This can be solved easily by well-known technical means. For example, the platform could post each assignment to a URI that would consist in the secure hash of the worker's perturbed profile. Each worker would then be able to access this URI based on a secure web browsing (*e.g.,* TOR[14]).

4 Evaluation

4.1 Experimental Settings

We evaluated the performance and quality reached by the `MWF`, `CWF`, and `TWF` weight functions. We implemented several comparison baselines: the `Hamming`

[14] https://www.torproject.org/

Fig. 4: Skills distribution of the `Normal` dataset over the `Perfect34` taxonomy

weight function, the `AWF` weight function, and a random matching. Performance is measured in terms of wall-clock time on a non-optimized implementation, and quality is measured according to the Definitions 2 and 3.

We use a synthetic taxonomy (`Perfect34`) that we generated from a perfect balanced tree of height 3 and branching factor 4, thus containing 85 nodes of which 64 are leaves.

Our experiments were performed over three synthetic datasets. (1) In the first dataset (`Normal`), the skills (workers and tasks) are randomly sampled from the leaves of the taxonomy following a normal distribution $\mathcal{N}(\sqrt{r}\cos\theta, \sqrt{r}|\sin\theta|)$ with θ randomly chosen in $[0,\ 2\pi)$ for each profile, and $r = 0.06$, corresponding to around 30% of skills per profile. The resulting distribution is given on Figure 4. (2) In the two other datasets, the skills of each worker's and task's profile are sampled uniformly at random in the leaves of the taxonomy at varying rates. We call `Bernoulli01` the dataset drawn from $\mathcal{B}(1, p = 0.01)$, and `Bernoulli1` the one from $\mathcal{B}(1, p = 0.1)$. All the assignments were made between 100 workers and 100 tasks.

The experiments presented in this paper were carried out using the Grid'5000 testbed[15]. All experiment were conducted on machines running Debian Jessie (64-bits) with an Intel(R) Xeon(R) E5-2680 v4 at 2.4 GHz and 16GB of RAM. The code was written in Python[16] 3.5.3 using NumPy[17] 1.12.0, SciPy[18] 0.18.1,

[15] https://www.grid5000.fr
[16] http://www.python.org
[17] http://www.numpy.org
[18] https://www.scipy.org

Weight function	Time (s)	Time complexity
`rand`	0.00099	$\mathcal{O}(1)$
`hamming`	0.02467	$\mathcal{O}(\lvert\mathcal{S}\rvert)$
`MWF`	0.01969	$\mathcal{O}(\lvert\mathcal{S}\rvert)$
`CWF`	0.30395	$\mathcal{O}(\lvert\mathcal{S}^T\rvert)$
`AWF`	1.99307	$\mathcal{O}(\lvert\mathcal{S}\rvert^2)$
`TWF`	2.96748	$\mathcal{O}(\lvert\mathcal{S}\rvert^2)$

Table 1: Wall-Clock Time of a Call to each Weight Function

NetworkX[19] 1.11, and JobLib[20] 0.10.4.dev0. Every experiment was run three times, unless stated otherwise.

4.2 Performance

Our approach does not generate any network overhead since a perturbed skills vectors has the same length as an original skills vector[21]. We consequently focus on the wall-clock time taken to compute the weight functions. Table 1 shows the average time among one thousand calls to each weight function, as well as their asymptotic complexities. The taxonomy is `Perfect34` and the dataset `Bernoulli1`. We assume pre-computed lowest-common-ancestors and shortest-paths, and that the distance used by `CWF` is linear in time. We observe that these times are insignificant with respect to typical computation times of the assignment algorithm.

4.3 Quality

Figure 5 shows the evolution of the two quality measures $\mathtt{q}_{\mathtt{rel}}$ and `fpa`, according to the bit flipping probability $\Pr_{flip}$, of the complete set of weight functions and on the random matching, on the synthetic taxonomy `Perfect34`. First, let consider the `Bernoulli` datasets. We observe globally that the weight functions are rather robust against the differentially-private perturbation. `TWF` exhibits a noticeable $\mathtt{q}_{\mathtt{rel}}$ measure equal to 0 on the two `Bernoulli` datasets. This is actually due to the 0 cost of the unperturbed assignment which makes $\mathtt{q}_{\mathtt{rel}}$ equal to 0 whatever the cost of the perturbed assignment. The later exhibited a value similar to the cost of `TWF` on the `Normal` dataset. Given its high relative quality, `Hamming` happens to be a robust metric. However, its `fpa` score is 0 (or close to) on the two `Bernoulli` datasets. This is not the case for the other metrics on `Bernoulli01`, especially for `MWF` and `AWF` and except `TWF`. Second, on the normal dataset the weight functions using the taxonomy (*i.e.,* `TWF`, `CWF`,

[19] https://networkx.github.io
[20] https://pythonhosted.org/joblib
[21] We note that though the length of a skills vector is unchanged, the Flip mechanism could increase the entropy, thus reducing compressing capabilities.

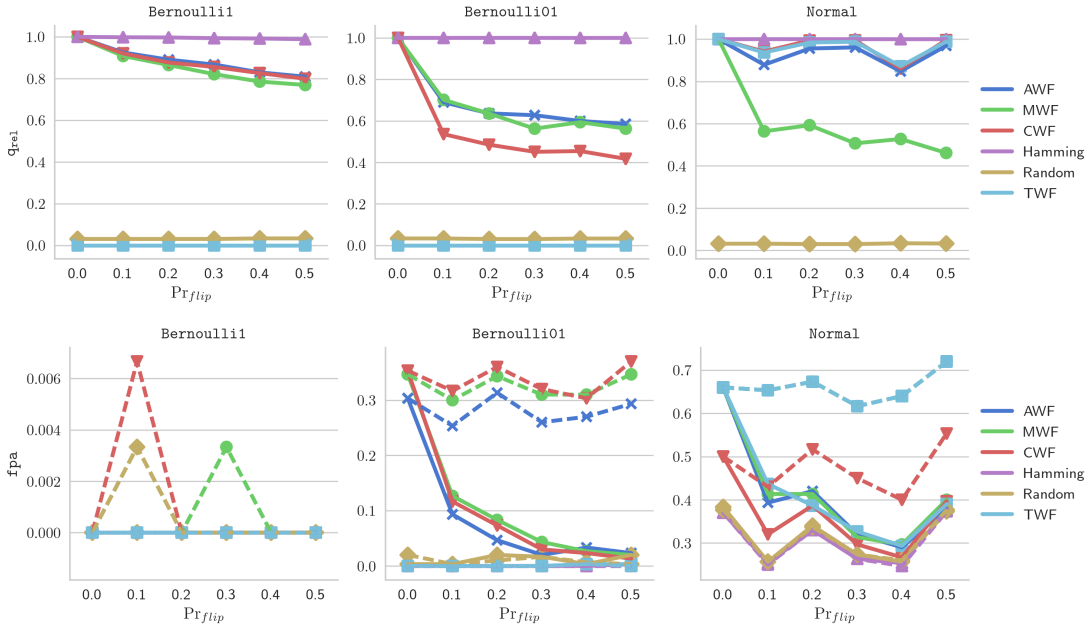

Fig. 5: $\mathtt{q_{rel}}$ and `fpa` for each weight function, with respect to $\Pr_{flip}$, for the `Perfect34` taxonomy. `fpa` of non-perturbated assignments is dashed

`AWF`) have a relative quality close to 1 robust and achieve a reasonnable, similar, `fpa` score.

5 Related Work

Privacy-Preserving Crowdsourcing A few recent studies have investigated privacy issues in crowdsourcing task assignments [17, 8], but they suffer from a strongly limited applicability either because they focus on a specific application domain (data type, crowdsourcing process), or because their cost is prohibitively high (rely on costly homomorphic encryption primitives). More precisely, first, differential privacy has been used recently to build a privacy-aware framework for spatial crowdsourcing [17]. This work enables a spatio-temporal task assignment performed according to the current geolocation of the workers but without disclosing it. It consists in computing a privacy-preserving spatial index over the set of workers' geolocations, and in routing each spatial task to the workers that are currently physically close to the task location. Their framework does not support the process and the data type that we consider (crowdsourcing based on skills). Second, a private task assignment protocol was proposed in [8] which produces an optimal task assignment while respecting the privacy of workers and of requesters. The protocol basically relies on the Paillier homomorphic cryptosystem [16] for protecting workers' profiles and the tasks. However, the overhead

due to the use of these cryptographic primitives make the method unable to cope with realistic cost requirements. Indeed, our theoretical run-time analysis shows that more than a century would be needed to compute an assignment between a hundred workers and a hundred tasks.

Other Close Problems The approach proposed in [9] aims at protecting the privacy of respondents during the surveys posted on a crowdsourcing platform. Given a survey and an history of perturbed responses, a challenge lies in forming the smallest group of respondents that will minimize the expected error according to their past history. Their problem is thus different from our assignment problem which results in the design and use of different families of algorithms. Similarly, the high-level goal of privacy-preserving recommender systems [15] is close to ours, but their problem is not the assignment problem (no bijection). The stable matching problem consists in computing the optimal matching between two sets such that each element in each set has its own matching preferences (*i.e.,* an ordered list of elements of the other set). The canonical real-life example is the matching between medical students and medical residency programs. A recent work considers the security issues of the stable matching problem [4] and proposes an elegant cryptographic solution for protecting the two sets to be matched. However, in addition to being a different problem, they suffer from a high cryptographic cost (*e.g.,* over 18 hours to complete a matching with around 35K participants and 30K residency slots).

Alternative Randomized Response Mechanisms Finally, randomized response mechanisms have been used in a variety of works but essentially as a local perturbation method for gathering statistics while providing strong privacy guarantees. For example, the recent work [20] proposes to use randomised response over bloom filters instead of raw bit vectors in order to enhance the quality of the final statistics when the same population is repeatedly surveyed. Our approach could use alternative randomized response schemes provided that it satisfies differential privacy and can be performed locally by each worker.

6 Conclusion

We have described in this paper a lightweight privacy-preserving approach to the problem of assigning tasks to workers. Our approach allows each worker to perturb her skill profile locally in order to satisfy the stringent differential privacy model without any need for additional communication or computation cost. We have proposed novel weight functions that can be easily plugged in traditional centralized assignment algorithms, and that are able to cope with the differentially private perturbation by leveraging the presence of a skill taxonomy. Experiments performed over a synthetic taxonomy and synthetic datasets show that our weight functions allow perturbed assignements to reach quality levels close to non-perturbed assignments. Future works include collecting a large-scale real-life skill dataset, and continue exploring the performance vs quality trade-off by designing alternative profile perturbation strategies (*e.g.,* collaborative perturbation protocols).

References

1. D. P. Bertsekas. A new algorithm for the assignment problem. *Mathematical Programming*, 21(1):152–171, 1981.
2. I.-J. Bienaymé. *Considérations à l'appui de la découverte de Laplace sur la loi de probabilité dans la méthode des moindres carrés.* Imprim. Mallet-Bachelier, 1853.
3. G. Blair, K. Imai, and Y.-Y. Zhou. Design and analysis of the randomized response technique. *Journal of the American Stat. Assoc.*, 110(511):1304–1319, 2015.
4. J. Doerner, D. Evans, and a. shelat. Secure stable matching at scale. In *Proc. of ACM CCS '16*, pages 1602–1613, 2016.
5. C. Dwork. Differential privacy. In *Proc. of ICALP '06*, pages 1–12, 2006.
6. C. Dwork. *Differential Privacy: A Survey of Results*, pages 1–19. 2008.
7. C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014.
8. H. Kajino. *Privacy-Preserving Crowdsourcing.* PhD thesis, Univ. Tokyo, 2016.
9. T. Kandappu, V. Sivaraman, A. Friedman, and R. Boreli. Loki: A privacy-conscious platform for crowdsourced surveys. In *COMSNETS '14*, pages 1–8, 2014.
10. S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith. What can we learn privately? In *Proc. of IEEE FOCS '08*, pages 531–540, 2008.
11. H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.
12. M. Lease, J. Hullman, J. P. Bigham, M. S. Bernstein, J. Kim, W. Lasecki, S. Bakhshi, T. Mitra, and R. C. Miller. Mechanical turk is not anonymous. *SSRN Electronic Journal*, 2013.
13. P. Mavridis, D. Gross-Amblard, and Z. Miklós. Using hierarchical skills for optimized task assignment in knowledge-intensive crowdsourcing. In *Proc. of WWW '16*, pages 843–853, 2016.
14. F. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In *Proc. of ACM SIGMOD '09*, pages 19–30, 2009.
15. F. McSherry and I. Mironov. Differentially private recommender systems. In *Proc. of ACM SIGKDD '09*, pages 627–636, 2009.
16. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT '99*, pages 223–238, 1999.
17. H. To, G. Ghinita, and C. Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing. *Proc. VLDB Endow.*, 7(10):919–930, June 2014.
18. S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Stat. Assoc.*, 60(309):63–69, 1965.
19. A. Waseda and R. Nojima. Analyzing randomized response mechanisms under differential privacy. In *ISC '16*, pages 271–282, 2016.
20. Úlfar Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proc. of ACM CCS '14*, 2014.