Private Approximation of Clustering and Vertex Cover*

Amos Beimel, Renen Hallak, and Kobbi Nissim

Department of Computer Science, Ben-Gurion University of the Negev

Abstract. Private approximation of search problems deals with finding approximate solutions to search problems while disclosing as little information as possible. The focus of this work is on private approximation of the vertex cover problem and two well studied clustering problems – k-center and k-median. Vertex cover was considered in [Beimel, Carmi, Nissim, and Weinreb, *STOC*, 2006] and we improve their infeasibility results. Clustering algorithms are frequently applied to sensitive data, and hence are of interest in the contexts of secure computation and private approximation. We show that these problems do not admit private approximations, or even approximation algorithms that leak significant number of bits. For the vertex cover problem we show a tight infeasibility result: every algorithm that $\rho(n)$-approximates vertex-cover must leak $\Omega(n/\rho(n))$ bits (where n is the number of vertices in the graph). For the clustering problems we prove that even approximation algorithms with a poor approximation ratio must leak $\Omega(n)$ bits (where n is the number of points in the instance). For these results we develop new proof techniques, which are more simple and intuitive than those in Beimel et al., and yet allow stronger infeasibility results. Our proofs rely on the hardness of the promise problem where a unique optimal solution exists [Valiant and Vazirani, *Theoretical Computer Science*, 1986], on the hardness of approximating witnesses for NP-hard problems ([Kumar and Sivakumar, *CCC*, 1999] and [Feige, Langberg, and Nissim, *APPROX*, 2000]), and on a simple random embedding of instances into bigger instances.

1 Introduction

In secure multiparty computation two or more parties wish to perform a computation over their joint data without leaking any other information. By the general feasibility results of [22,8,2], this task is well defined and completely solved for polynomial time computable functions. When what the parties wish to compute is *not a function*, or infeasible to compute (or both) one cannot directly apply the feasibility results, and special care has to be taken in choosing the function that is computed securely, as the outcome of the secure computation may

* Research partially supported by the Israel Science Foundation (grant No. 860/06), and by the Frankel Center for Computer Science. Research partly done when the first and third authors were at the Institute for Pure and Applied Mathematics, UCLA.

leak information. We deal with such problems – vertex-cover and clustering that are NP-complete problems – and check the consequences of choosing to compute *private approximations* for these search problems, i.e., approximation algorithms that do not leak more information than the collection of solutions for the specific instance.

The notion of private approximation was first put forward and researched in the context of approximating *functions* [6,10], and was recently extended to search problems [1]. These works also consider relaxations of private approximations, which allow for a bounded leakage. The research of private approximations yielded mixed results: (i) private approximation algorithms or algorithms that leak very little were presented for well studied problems [6,10,7,15,13,1], but (ii) it was shown that some natural functions do not admit private approximations, unless some (small) leakage is allowed [10]; and some search problems do not even admit approximation algorithms with significant leakage [1]. We continue the later line of research and prove that vertex-cover and two clustering problems – k-center and k-median – do not admit private approximation algorithms, or even approximation algorithms that leak significant number of bits.

1.1 Previous Works

Feigenbaum et al. [6] noted that an approximation to a function may reveal information on the instance that is not revealed by the exact (or optimal) function outcome. Hence, they formulated, , via the simulation paradigm, a notion of private approximations that prevents exactly this leakage. Their definition implies that if applied to instances x, y such that $f(x) = f(y)$, the outcome of an approximation algorithm $\hat{f}(x), \hat{f}(y)$ are indistinguishable. Under their definition of private approximations, Feigenbaum et al. provided a protocol for approximating the Hamming distance of two n-bit strings with communication complexity $\tilde{O}(\sqrt{n})$, and polynomial solutions for approximating the permanent and other natural #P problems. Subsequent work on private approximations improved the communication complexity for the Hamming distance to polylog(n) [13]. Other works on private approximations for specific functions include [15,7].

Attempts to constructs private approximations of the objective functions of certain NP-complete problems were unsuccessful. This phenomenon was explained by Halevi, Krauthgamer, Kushilevitz, and Nissim [10] proving strong inapproximability results for computing the size of a minimum vertex cover even within approximation ratio $n^{1-\epsilon}$. They, therefore, presented a relaxation, allowing the leakage of a deterministic predicate of the input. Fortunately, this slight compromise in privacy allows fairly good approximations for any problem that admits a good deterministic approximation. For example, minimum vertex cover may be approximated within a ratio of 4 leaking just one bit of approximation.

Recently, Beimel, Carmi, Nissim, and Weinreb [1] extended the privacy requirement of [6] from *functions* to *search problems*, giving a (seemingly) lenient definition which only allows leaking whatever is implied by the set of all exact solutions to the problem. A little more formally, if applied to instances x, y that share *exactly* the same set of (optimal) solutions, the outcome of the approximation algorithm $\mathcal{A}(x)$ on x should be indistinguishable from $\mathcal{A}(y)$. They showed that even under this definition it is not feasible to privately approximate the search problems of vertex-cover and 3SAT. Adopting the relaxation of [10] to the context of private search, Beimel et al. showed for max exact 3SAT an approximation algorithm with a near optimal approximation ratio of $7/8 - \epsilon$ that leaks only $O(\log \log n)$ bits. For vertex-cover, the improvement is more modest – there exists an approximation algorithm within ratio $\rho(n)$ that leaks $\ell(n)$ bits where $\rho(n) \cdot \ell(n) = 2n$. On the other hand, they proved that an algorithm for vertex-cover that leaks $O(\log n)$ bits cannot achieve n^ϵ approximation. We close this gap up to constant factors. A different relaxation of private approximation was presented in the context of near neighbor search by Indyk and Woodruff [13], and we refer to a generalization of this relaxation in Section 4.

1.2 Our Contributions

The main part of this work investigates how the notion of private approximations and its variants combine with well studied NP-complete *search* problems – vertex-cover, k-center, and k-median. We give strong infeasibility results for these problems that hold with respect to a more lenient privacy definition than in [1] – that only requires that $\mathcal{A}(x)$ is indistinguishable from $\mathcal{A}(y)$ on instances x, y that have the same *unique* solution. To prove our results, we introduce new strong techniques for proving the infeasibility of private approximations, even with many bits of leakage.

Vertex Cover. As noted above, the feasibility of private approximation of vertex-cover was researched in [1]. Their analysis left an exponential gap between the infeasibility and feasibility results. We close this gap, and show that, unless RP = NP, any approximation algorithm that leaks at most $\ell(n)$ bits of information and is within approximation ratio $\rho(n)$ satisfies $\rho(n) \cdot \ell(n) = \Omega(n)$. This result is tight (up to constant factors) by a result described in [1]: for every constant $\epsilon > 0$, there is an $n^{1-\epsilon}$-approximation algorithm for vertex-cover that leaks $2n^\epsilon$ bits.

Clustering. Clustering is the problem of partitioning n data points into disjoint sets in order to minimize a cost objective related to the distances within each point set. Variants of clustering are the focus of much research in data mining and machine learning as well as pattern recognition, image analysis, and bioinformatics. We consider two variants: (i) k-center, where the cost of a clustering is taken to be the longest distance of a point from its closest center; and (ii) k-median, where the cost is taken to be the average distance of points from their closest centers. Both problems are NP-complete [12,14,18]. Furthermore, we consider two versions of each problem, the one outputting the indices of the centers and the second outputting the coordinates of the solutions. For private algorithms these two versions are not equivalent since different information can be learned from the output.

We prove that, unless RP = NP, every approximation algorithm for the indices version of these problems must leak $\Omega(n)$ bits even if its approximation ratio as poor as $2^{\mathrm{poly}(n)}$. As there is a 2-approximation algorithm that leaks at most n bits (the incidence vector of the set of centers), our result is tight up to a constant factor. Similar results are proved in the full version of the paper for the coordinate version of these problems (using a "perturbable" property of the metric).

Trying to get around the impossibility results, we examine a generalization of a privacy definition by Indyk and Woodruff [13], originally presented in the context of near neighbor search. In the modified definition, the approximation algorithm is allowed to leak the set of η-approximated solutions to an instance for a given η. We consider the coordinate version of k-center, and show that there exists a private 2-approximation under this definition for every $\eta \geq 2$, and there is no approximation algorithm under this definition when $\eta < 2$.

New Techniques. The basic idea of our infeasibility proofs is to assume that there exists an efficient private approximation algorithm $\mathcal{A}$ for some NP-complete problem, and use this algorithm to efficiently find an *optimal* solution of the problem contradicting the NP-hardness of the problem. Specifically, in our proofs we take an instance x of the NP-complete problem, transform it to a new instance x', execute $y' \leftarrow \mathcal{A}(x')$ once getting an *approximate solution* for x', and then efficiently reconstruct from y' an *optimal solution* for x. Thus, we construct a Karp-reduction from the original NP-complete problem to the private approximation version of the problem. This should be compared to the reduction in [1] which used many calls to $\mathcal{A}$, where the inputs to $\mathcal{A}$ are chosen adaptively, according to the previous answers of $\mathcal{A}$.

Our techniques differ significantly from those of [1], and are very intuitive and rather simple. The main difference is that we deal with the promise versions of vertex cover and clustering, where a *unique* optimal solution exists. These problems are also NP-hard under randomized reductions [21]. Analyzing how a private approximation algorithms operate on instances of the promise problem, we clearly identify a source for hardness in an attempt to create such an algorithm – it, essentially, has to output the optimal solution. Furthermore, proving the infeasibility result for instances of the unique problems shows that hardness of private approximation stems from instances we are trying to approximate a "function" – given an instance the function returns its unique optimal solution. Thus, our impossibility results are for inputs with unique solutions where the privacy requirement is even more minimal than the definition of [1].

To get our strongest impossibility results, we use the results of Kumar and Sivakumar [16] and Feige, Langberg, and Nissim [5] that, for many NP-complete problems, it is NP-hard to approximate the witnesses (that is, viewing a witness and an approximation as sets, we require that their symmetric difference is small). These results embed a redundant encoding of the optimal solution, so that seeing a "noisy" version of the optimal solution allows recovering it. In our infeasibility proofs, we assume that there exists an approximation algorithm $\mathcal{A}$ for some unique problem, and use this algorithm to find a solution close to the optimal solution. Thus, the NP-hardness results of [16,5] imply that such efficient algorithm $\mathcal{A}$ cannot exist.

Our last technique is a simple random embedding of an instance into a bigger instance. Let us demonstrate this idea for the unique-vertex-cover problem. In this case, we take a graph, add polynomially many isolated vertices, and then randomly permute the names of the vertices. We assume that there exists a private approximation algorithm $\mathcal{A}$ for vertex-cover and we execute $\mathcal{A}$ on the bigger instance. We show that, with high probability, the only vertices from the original graph that appear in the output of $\mathcal{A}$ are the vertices of the unique vertex cover of the original graph. The intuition behind this phenomenon is that, by the privacy requirement, $\mathcal{A}$ has to give the same answer for many instances generated by different random permutations of the names, hence, if a vertex is in the answer of $\mathcal{A}$, then with high probability it corresponds to an isolated vertex.

Organization. Section 2 contains the main definitions used in this paper and essential background. Section 3 includes our impossibility result for almost private algorithms for the index version of k-center, based on the hardness of unique-k-center. Section 4 discusses an alternative definition of private approximation of the coordinate version of k-center, and contains possibility and impossibility results for this definition. Section 5 describes our impossibility result for almost private algorithms for vertex-cover. Finally, Section 6 discusses some questions arising from our work.

2 Preliminaries

In this section we give definitions and background needed for this paper. We start with the definitions of private search algorithms from [1]. Thereafter, we discuss the problems we focus on: the clustering problems – k-center and k-median – and vertex cover. We then define a simple property of the underlying metrics that will allow us to present our results in a metric independent manner. Finally, we discuss two tools we use to prove infeasibility results: (1) hardness of unique problems and parsimonious reductions, and (2) error correcting reductions.

2.1 Private Approximation of Search Problems

Beimel et al. [1] define the privacy of search algorithms with respect to some underlying privacy structure $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ that is an equivalence relation on instances. The notation $x \equiv_{\mathcal{R}} y$ denotes $\langle x, y \rangle \in \mathcal{R}$. The equivalence relation determines which instances should not be told apart by a private search algorithm $\mathcal{A}$:

Definition 1 (Private Search Algorithm [1]). *Let $\mathcal{R}$ be a privacy structure. A probabilistic polynomial time algorithm $\mathcal{A}$ is* private *with respect to $\mathcal{R}$ if for every polynomial-time algorithm $\mathcal{D}$ and for every positive polynomial $p(\cdot)$, there* *exists some $n_0 \in \mathbb{N}$ such that for every $x, y \in \{0,1\}^*$ such that $x \equiv_{\mathcal{R}} y$ and $|x| = |y| \geq n_0$*

$$\Big| \Pr[\mathcal{D}(\mathcal{A}(x), x, y) = 1] - \Pr[\mathcal{D}(\mathcal{A}(y), x, y) = 1] \Big| \leq \frac{1}{p(|x|)},$$

where the probabilities are taken over the random choices of $\mathcal{A}$ and $\mathcal{D}$.

For every search problem, a related privacy structure is defined in [1], where two inputs are equivalent if they have the same set of optimal solutions. In Section 2.2 we give the specific definitions for the problems we consider.

We will also use the relaxed version of Definition 1 that allows a (bounded) leakage. An equivalence relation $\mathcal{R}'$ is said to ℓ-refine an equivalence relation $\mathcal{R}$ if $\mathcal{R}' \subseteq \mathcal{R}$ and every equivalence class of $\mathcal{R}$ is a union of at most 2^ℓ equivalence classes of $\mathcal{R}'$.

Definition 2 ([1]). *Let $\mathcal{R}$ be a privacy structure. A probabilistic polynomial time algorithm $\mathcal{A}$ leaks at most ℓ bits with respect to $\mathcal{R}$ if there exists a privacy structure $\mathcal{R}'$ such that (i) $\mathcal{R}'$ is a ℓ-refinement of $\mathcal{R}$, and (ii) $\mathcal{A}$ is private with respect to $\mathcal{R}'$.*

2.2 *k*-center and *k*-median Clustering

The k-center and k-median clustering problems are well researched problems, both known to be NP-complete [12,14,18]. In both problems, the input is a collection P of points in some metric space and a parameter c. The output is a collection of c of the points in P – the cluster centers – specified by their indices or by their coordinates. The partition into clusters follows by assigning each point to its closest center (breaking ties arbitrarily). The difference between k-center and k-median is in the cost function: in k-center the cost is taken to be the maximum distance of a point in P from its nearest center; in k-median it is taken to be the average distance of points from their closest centers. For private algorithms, the choice of outputting indices or coordinates may be significant (different information can be learned from each), and hence we define two versions of each problem.

Definition 3 (k-center – outputting indices (k-center-I)). *Given a set $P = \{p_1, \ldots, p_n\}$ of n points in a metric space and a parameter c, return the indices of c cluster centers $I = \{i_1, \ldots, i_c\}$ that minimize the maximum cluster radius.*

Definition 4 (k-center – outputting coordinates (k-center-C)). *Given a set $P = \{p_1, p_2, \ldots, p_n\}$ of n points in a metric space and a parameter c, return the coordinates of c cluster centers $C = \{p_{i_1}, \ldots, p_{i_c}\}$ that minimize the maximum cluster radius ($C \subseteq P$).*[1]

[1] We do not consider versions of the problem where the centers do not need to be points in P.

The k-median-I and k-median-C problems are defined analogously.

Theorem 1 ([12,14,18]). *In a general metric space,* k-center *(*k-median*) is* NP*-hard. Furthermore, the problem of finding a* $(2-\epsilon)$*-approximation of* k*-center in a general metric space is* NP*-hard for every* $\epsilon > 0$*.*

Proof (sketch): The reduction is from dominating set. Given a graph $G = (V, E)$, transform each vertex $v \in V$ to a point $p \in P$. For every two points $p_1, p_2 \in P$ let $\text{dist}(p_1, p_2) = 1$ if $(v_1, v_2) \in E$, otherwise $\text{dist}(p_1, p_2) = 2$. As the distances are 1 and 2, they satisfy the triangle inequality. There is a dominating set of size c in G iff there is a k-center clustering of size c and cost 1 (k-median clustering of cost $\frac{n-c}{n}$) in P. Furthermore, every solution to k-center with cost less than 2 in the constructed instance has cost 1, which implies the hardness of $(2-\epsilon)$-approximation for k-center. $\square$

There is a greedy 2-approximation algorithm for k-center [9,11]: select a first center arbitrarily, and iteratively selects the other $c-1$ points each time maximizing the distance to the previously selected centers. We will make use of the above reduction, as well as the 2-approximation algorithm for this problem, in the sequel.

We next define the privacy structures related to k-center. Only instances $(P_1, c_1), (P_2, c_2)$ were $|P_1| = |P_2|$ and $c_1 = c_2$ are equivalent, provided they satisfy the following conditions:

Definition 5. *Let* P_1, P_2 *be sets of* n *points and* $c < n$ *a parameter determining the number of cluster centers.*

- *Instances* (P_1, c) *and* (P_2, c) *are equivalent under the relation* $\mathcal{R}_{k\text{-center-I}}$ *if for every set* $I = \{i_1, \ldots, i_c\}$ *of* c *point indices,* I *minimizes the maximum cluster radius for* (P_1, c) *iff it minimizes the maximum cluster radius for* (P_2, c)*.*
- *Instances* (P_1, c) *and* (P_2, c) *are equivalent under the relation* $\mathcal{R}_{k\text{-center-C}}$ *if (i) for every set* $C \subseteq P_1$ *of* c *points, if* C *minimizes the maximum cluster radius for* (P_1, c) *then* $C \subseteq P_2$ *and it minimizes the maximum cluster radius for* (P_2, c)*; and similarly (ii) for every set* $C \subseteq P_2$ *of* c *points, if* C *minimizes the maximum cluster radius for* (P_2, c) *then* $C \subseteq P_1$ *and it minimizes the maximum cluster radius for* (P_1, c)

Definition 6 (Private Approximation of k-center). *A randomized algorithm* $\mathcal{A}$ *is a* private $\rho(n)$-approximation *algorithm for* k-center-I *(respectively* k-center-C*) if: (i) the algorithm* $\mathcal{A}$ *is a* $\rho(n)$*-approximation algorithm for* k-center*, that is, for every instance* (P, c) *with* n *points, it returns a solution – a set of* c *points – such that the expected cluster radius of the solution is at most* $\rho(n)$ *times the radius of the optimal solution of* (P, c)*. (ii)* $\mathcal{A}$ *is private with respect to* $\mathcal{R}_{k\text{-center-I}}$ *(respectively* k-center-C*).*

The definitions for vertex-cover are analogous and can be found in [1].

2.3 Distance Metric Spaces

In the infeasibility results for clustering problems we use a simple property of the metric spaces, which we state below. This allows us to keep the results general and metric independent. One should be aware that clustering problems may have varying degrees of difficulty depending on the underlying metric used. Our impossibility results will show that unique-k-center and unique-k-median may be exactly solved in *randomized polynomial time* if private algorithms for these problems exist. When using metric spaces for which the problems are NP-hard, this implies RP = NP.

The property states that given a collection of points, it is possible to add to it new points that are "far away":

Definition 7 (Expandable Metric). *Let $\mathcal{M}$ be a family of metric spaces. A family of metric spaces $\mathcal{M}$ is (ρ, m)-*expandable *if there exists an algorithm* Expand *that given a metric $M = \langle P, \mathrm{dist} \rangle \in \mathcal{M}$, where $P = \{p_1, \ldots, p_n\}$, runs in time polynomial in n, m, and the description of M, and outputs a metric $M' = \langle P', \mathrm{dist}' \rangle \in \mathcal{M}$, where $P' = \{p_1, \ldots, p_n, p_{n+1}, \ldots, p_{n+m}\}$, such that*

- $\mathrm{dist}'(p_i, p_j) = \mathrm{dist}(p_i, p_j)$ *for every $i, j \in [n]$, and*
- $\mathrm{dist}'(p_i, p_j) \geq \rho d$ *for all $n < i \leq n + m$ and $1 \leq j < i$, where $d = \max_{i,j \in [n]}(\mathrm{dist}(p_i, p_j))$ is the maximum distance within the original n points.*

General Metric Spaces. Given a connected undirected graph $G = (V, E)$ where every edge $e \in E$ has a positive length $w(e)$, define the metric induced by G whose points are the vertices and $\mathrm{dist}_G(u, v)$ is the length of the shortest path in G between u and v. The family $\mathcal{M}$ of general metric spaces is the family of all metric spaces induced by graphs. This family is expandable: Given a graph G, we construct a new graph G' by adding to G a path of m new vertices connected to an arbitrary vertex, where the length of every new edge is $\rho(n) \cdot d$. The metric induced by G' is the desired expansion of the metric induced by G. The expansion algorithm is polynomial when $\rho(n)$ is bounded by $2^{\mathrm{poly}(n)}$.

Observation 1. *Let $\rho(n) = 2^{\mathrm{poly}(n)}$. The family of general metric spaces is $(\rho(n), m)$-expandable for every m.*

Similarly, the family of metric spaces induced by a finite set of points in the plain with Euclidean distance is expandable.

2.4 Parsimonious Reductions and Unique Problems

Parsimonious reductions are reductions that preserve the number of solutions. It was observed that among the well known NP-complete problems, such reductions can be found [3,19,20]. Indeed, one can easily show that such reductions also exist for our problems:

Lemma 1. *SAT and 3-SAT are parsimoniously reducible to the vertex-cover, k-center, and k-median problems (the general metric version).*

The existence of such parsimonious reductions allows us to base our negative results on a promise version of the problems – where only a unique optimal solution exists. We use the results of Valiant and Vazirani [21] that the promise version unique-SAT is NP-hard under randomized reductions. Therefore, if there exists a parsimonious reduction from SAT to an NP-complete (search) problem $\mathcal{S}$, then its promise version unique-$\mathcal{S}$ is NP-hard under randomized reductions.

Corollary 1. Vertex-cover*,* unique-k-center*, and* unique-k-median *(general metric version) are* NP*-hard under randomized reductions.*

2.5 Error Correcting Reductions

An important tool in our proofs are error correcting reductions – reductions that encode, in a redundant manner, the witness for one NP-complete problem inside the witness for another. Such reductions were shown by Kumar and Sivakumar [16] and Feige, Langberg, and Nissim [5] – proving that for certain NP-complete problems it is hard to approximate witnesses (that is, when viewed as sets, the symmetric difference between the approximation and a witness is small). For example, such result is proved in [5] for vertex-cover. We observe that the proof in [5] applies to unique-vertex-cover and we present a similar result for unique-k-center and unique-k-median. We start by describing the result of [5] for unique-vertex-cover.

Definition 8 (Close to a minimum vertex cover). *A set S is δ-close to a minimum vertex cover of G if there exists a minimum vertex cover C of G such that $|S \triangle C| \leq (1-\delta)n$.*

Theorem 2 ([21,5]). *If* RP $\neq$ NP*, then for every constant $\delta > 1/2$ there is no efficient algorithm that, on input a graph G and an integer t where G has a unique vertex cover of size t, returns a set S that is δ-close to the minimum vertex cover of G.*

We next describe the result for unique-k-center.

Definition 9 (Close to an optimal solution of unique-k-center**).** *A set S is δ-close an optimal solution of an instance (P, c) of* unique-k-center *if there exists an optimal solution I of (P, c) such that $|S \triangle I| \leq (1-\delta)n$.*

Theorem 3. *If* RP $\neq$ NP*, then, for every constant $\delta > 2/3$, there is no efficient algorithm that for every instance (P, c) of* unique-k-center *finds a set δ-close to the optimal solution of (P, c). The same result holds for instances of* unique-k-median*.*

The proof technique of Theorem 3 is similar to the proofs in [5]. The proof is described in the full version of this paper.

3 Infeasibility of Almost Private Approximation of Clustering

In this section, we prove that if RP $\neq$ NP, then every approximation algorithm for the clustering problems is not private (and, in fact, must leak $\Omega(n)$ bits). We will give a complete treatment for k-center-I (assuming the underlying metric is expandable according to Definition 7). The modifications needed for k-median-I are small. The proof for k-center-C and k-median-C are different and use a "perturbable" property of the metric. The proofs for the 3 latter problems appear in the full version of this paper. We will start our proof for k-center-I by describing the infeasibility result for private algorithms, and then we consider deterministic almost private algorithms. The infeasibility result for randomized almost private algorithms appears in the full version of this paper.

3.1 Infeasibility of Private Approximation of Clustering Problems

In this section, we demonstrate that the existence of a private approximation algorithm for k-center-I implies that unique-k-center is in RP. Using the hardness of the promise version unique-k-center, we get our infeasibility result.

We will now show that any private $\rho(n)$-approximation algorithm must essentially return all the points in the unique solution of an instance. We use the fact that the underlying metric is $(2n \cdot \rho(n+1), 1)$-*expandable*. Given an instance $(P, c) = (\{p_1, \ldots, p_n\}, c)$ for k-center-I we use Algorithm Expand with parameters $(2n \cdot \rho(n+1), 1)$ to create an instance $(P', c+1)$ by adding the point p^∞ returned by Expand, i.e. $p_{n+1} = p^\infty$ and $\mathrm{dist}'(p_i, p^\infty) \geq \rho(n+1) \cdot d$. Any optimal solution I' for $(P', c+1)$ includes the new point p^∞ (if $p^\infty \notin I'$ then this solution's cost is at least $2n \cdot \rho(n+1) \cdot d$ whereas if $p^\infty \in I'$ the cost is at most d). Hence, the unique optimal solution I' consists of the optimal solution I for (P, c) plus the index $n+1$ of the point p^∞.

Lemma 2. *Let $\mathcal{A}$ be a private $\rho(n)$-approximation algorithm for* k-center-I*, let (P, c) be an instance of* k-center-I *and construct $(P', c+1)$ as above. Then*

$$\Pr[\mathcal{A}(P', c+1) \textit{ returns the indices of all critical points of } (P, c)] \geq 1/3 \,.$$

The probability is taken over the random coins of algorithm $\mathcal{A}$.

Proof. Let $p_{i_1}, \ldots, p_{i_c}$ be the points of the unique optimal solution of (P, c) (hence $p_{i_1}, \ldots, p_{i_c}, p_{n+1}$ are the points of the unique optimal solution of $(P', c+1)$). Consider an instance $(P'', c+1)$ where P'' is identical to P', except for the points p_{i_1} and p^∞ whose indices (i_i and $n+1$) are swapped.[2] As both p_{i_1} and p^∞ are the optimal solution in P', swapping them does not change the optimal solution, and hence $(P'', c+1) \equiv_{\mathcal{R}_{k\text{-center-I}}} (P', c+1)$.

[2] Note that while P' can be efficiently constructed from P, the construction of P'' is only a thought experiment.

Let $\tilde{I}'$ and $\tilde{I}''$ denote the random variables $\mathcal{A}(P', c+1)$ and $\mathcal{A}(P'', c+1)$ respectively. Note that the optimal cost of $(P'', c+1)$ is bounded by d. Whereas if $i_1 \notin \tilde{I}''$ we get a clustering cost of $2n \cdot \rho(n+1) \cdot d$. Hence, if $\Pr[i_1 \notin \tilde{I}''] > 1/(2n)$ algorithm $\mathcal{A}$ cannot maintain an approximation ratio of $\rho(n+1)$. This implies that $\Pr[i_1 \notin \tilde{I}'] < 2/(3n)$, otherwise, it is easy to construct a polynomial time procedure that would distinguish $(\tilde{I}', P', P'')$ from $(\tilde{I}'', P', P'')$ with advantage $\Omega(1/n)$. A similar argument holds for indices $i_2, \ldots, i_c$.

To conclude the proof, we use the union bound and get that $\Pr[\{i_1, \ldots, i_m\} \subset \tilde{I}'] \geq 1 - 2c/3n \geq 1/3$. □

We now get our infeasibility result:

Theorem 4. *Let $\rho(n) \leq 2^{\mathrm{poly}(n)}$. The* k-center-I *problem does not admit a polynomial time private $\rho(n)$-approximation unless* unique-k-center *can be solved in probabilistic polynomial time.*

Proof. Let $\mathcal{A}$ be a polynomial time private $\rho(n)$-approximation for k-center-I. Let $(P, c) = (\{p_1, \ldots, p_n\}, c)$ be an instance of unique-k-center and let I be the indices of the centers in its unique solution. Construct the instance $(P', c+1)$ as above by adding the point $p_{n+1} = p^\infty$. As $\rho(n) \leq 2^{\mathrm{poly}(n)}$, constructing P' using Algorithm Expand is efficient. By Lemma 2, $\mathcal{A}(P')$ includes every index in I with probability at least 1/3. With high probability, $\mathcal{A}(P', c+1)$ contains exactly c points from P, and the set $\mathcal{A}(P') \setminus \{n+1\}$ is the unique optimal solution for (P, c). □

Combining Theorem 4 with Corollary 1 we get:

Corollary 2. *Let $\rho(n) \leq 2^{\mathrm{poly}(n)}$. The* k-center-I *problem (general metric version) cannot be privately $\rho(n)$-approximated in polynomial time unless* RP $\neq$ NP.

3.2 Infeasibility of Deterministic Approximation of Clustering Problems that Leaks Many Bits

In this section we prove that even if RP $\neq$ NP, then for every $\rho(n) \leq 2^{\mathrm{poly}(n)}$ there is no efficient *deterministic* $\rho(n)$-approximation algorithm of k-center-I that leaks $0.015n$ bits (as in Definition 2).[3] As in the previous section, we assume the underlying distance metric is expandable. To prove the infeasibility of almost private approximation of k-center-I, we assume towards contradiction that there exists an efficient deterministic $\rho(n)$-approximation algorithm $\mathcal{A}$ that leaks $0.015n$ bits. We use this algorithm to find a set close to the solution of a unique-k-center instance.

In the proof of the infeasibility result for private algorithms, described in Section 3.1, we started with an instance P of unique-k-center and generated a new instance P' by adding to P a "far" point. We considered an instance P'' that is equivalent to P' and argued that, since the instances are equivalent, a deterministic private algorithm must return the same output on the two instances.

[3] Throughout this paper, constants are shamelessly not optimized.

For almost private algorithms, we cannot use the same proof. Although the instances P' and P'' are equivalent, even an algorithm that leaks one bit can give different answers on P' and P''.

The first idea to overcome this problem is to add linearly many new "far" points (using Algorithm Expand). Thus, any deterministic approximation algorithm must return all "far" points and a subset of the original points. However, there is no guarantee that this subset is the optimal solution to the original instance. The second idea is using a random renaming of the indices of the instance. We will prove that with high probability (over the random choice of the renaming), the output of the almost private algorithm is close to the optimal solution of unique-k-center. This contradicts the NP-hardness, described in Section 2.5, of finding a set close to the exact solution for unique-k-center instances.

We next formally define the construction of adding "far" points and permuting the names. Given an instance (P, c) of unique-k-center with distance function dist, we use Algorithm Expand with parameters $(2 \cdot \rho(10n), 9n)$ to create an instance $(P', 9n + c)$ with distance function dist' by adding $9n$ "far" points. Let $N \stackrel{\text{def}}{=} 10n$ be the number of points in P' and $c' \stackrel{\text{def}}{=} c + 9n$. We next choose a permutation $\pi : [N] \to [N]$ to create a new instance $(P_\pi, 9n + c)$ with distance function dist_π, where $\text{dist}_\pi(p_{\pi(i)}, p_{\pi(j)}) \stackrel{\text{def}}{=} \text{dist}'(p_i, p_j)$.

We start with some notation. Let I be the the set of indices of the points in the unique optimal solution for (P, c) and $S \stackrel{\text{def}}{=} [n] \setminus I$ (that is, S is the set of indices of the points in the original instance P not in the optimal solution). Note that $|I| = c$ and $|S| = n - c$. For any set $A \subseteq [N]$, we denote $\pi(A) \stackrel{\text{def}}{=} \{\pi(i) : i \in A\}$. The construction of P_π and the sets S and I are illustrated in Fig. 1.

It is easy to see that an optimal solution I_π for (P_π, c') includes the $9n$ "far" points, that is, $\{p_{\pi(i)} : n + 1 \leq i \leq 10n\}$ (if not, then this solution's cost is at least $2 \cdot \rho(N) \cdot d$ whereas if $\{\pi(n + 1), \ldots, \pi(10n)\} \subset I_\pi$ the cost is at most d). Thus, I_π contains exactly c points from $\{p_{\pi(i)} : 1 \leq i \leq n\}$ which must be $\pi(I)$. That is, the unique optimal solution I_π of (P_π, c') consists of the indices in $[N] \setminus \pi(S)$.

Observation 2. *Let π_1, π_2 be two permutations such that $\pi_1(S) = \pi_2(S)$. Then, $(P_{\pi_1}, c') \equiv_{\mathcal{R}_{k\text{-center-I}}} (P_{\pi_2}, c')$.*

In Fig. 2 we describe Algorithm Close to Unique k-Center that finds a set close to the unique minimum solution of an instance of unique-k-center assuming the existence of a deterministic $\rho(N)$-approximation algorithm $\mathcal{A}$ for k-center-I that leaks $0.015N$-bits. Notice that in this algorithm we execute the approximation algorithm $\mathcal{A}$ on (P_π, c') – an instance with $N = 10n$ points – hence the approximation ratio of $\mathcal{A}$ (and its leakage) is a function of N.

We next prove that, with high probability, Algorithm Close to Unique k-Center returns a set that is close to the optimal solution. In the *analysis*, we partition the set of permutations $\pi : [N] \to [N]$ to disjoint subsets. We prove that in every subset, with high probability, Algorithm Close to Unique k-Center returns a set that is close to the optimal solution, provided that it

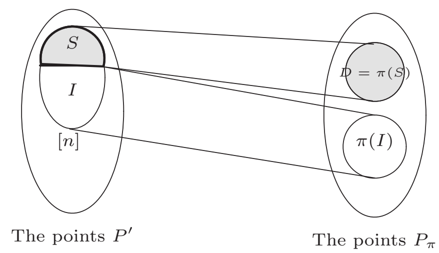

Fig. 1. The construction of P_π.

Algorithm Close to Unique k-Center:
Input: An instance $(P = \{p_1, \ldots, p_n\}, c)$ and an integer t.
Promise: (P, c) has a unique set of c cluster centers with maximum cluster radius at most t.
Output: A set 0.7-close to the unique set of c cluster centers with maximum cluster radius at most t.

1. Use algorithm Expand with parameters $(2\cdot\rho(10n), 9n)$ to create a set of points $P' = \{p_1, \ldots, p_n, p_{n+1}, \ldots, p_{10n}\}$.
2. Choose a permutation $\pi : [N] \to [N]$ uniformly at random and construct P_π.
3. Let $B \leftarrow \mathcal{A}(P_\pi, c + 9n)$ and $B^{-1} \leftarrow \{i \in [n] : \pi(i) \in B\}$.
4. Return B^{-1}.

Fig. 2. An algorithm that finds a set 0.7-close to the unique minimum solution of an instance of unique-k-center assuming that $\mathcal{A}$ is an almost private approximation algorithm for k-center-I.

chose a permutation in the subset. Specifically, for every $D \subset [N]$, we consider the subset of the permutations π such that $\pi(S) = D$.

In the rest of the proof we fix an instance (P, c) with a unique optimal solution I and define $S \stackrel{\text{def}}{=} [n] \setminus I$. Furthermore, we fix a set $D \subset [N]$ such that $|D| = |S|$ and consider only permutations such that $\pi(S) = D$. (The algorithm does not to need know S and D; these sets are used for the analysis.) We prove in Lemma 4 that with high probability $[N] \setminus \mathcal{A}(P_\pi, c')$ is close to D, and we show in Lemma 3 that in this case Algorithm Close to Unique k-Center succeeds.

Lemma 3. *Let B be a set such that $|B \cap D| \le 0.15n$ and π is a permutation such that $\mathcal{A}(P_\pi, c') = B$. Then, Algorithm* Close to Unique k-Center *returns a set 0.7-close to I when it chooses the permutation π in Step (2).*

Proof. When choosing π, Algorithm Close to Unique k-Center returns the set

$$\begin{aligned} B^{-1} &= \{i \in [n] : \pi(i) \in B\} = \{i \in I : \pi(i) \in B\} \cup \{i \in S : \pi(i) \in B\} \\ &= \{i \in I : \pi(i) \in B\} \cup \{i : \pi(i) \in B \cap D\}. \end{aligned}$$

Thus, $|B^{-1} \setminus I| = |B \cap D| \leq 0.15n$. As $|B^{-1}| = |I|$, we get $|I \setminus B^{-1}| = |B^{-1} \setminus I| \leq 0.15n$. Therefore, $|B^{-1} \triangle I| \leq 0.3n$, and B^{-1} is 0.7-close to I. $\square$

Lemma 4. *Let* $pr \stackrel{\text{def}}{=} \Pr[\,|\mathcal{A}(P_\pi, c') \cap D| \leq 0.15n\,]$, *where the probability is taken over the uniform choice of* π *subject to* $\pi(S) = D$. *Then,* $pr \geq 3/4$.

Proof. We prove that if $\text{pr} < 3/4$, there is a permutation π such that $\mathcal{A}$ does not $\rho(N)$-approximate k-center-I on (P_π, c'), in a contradiction to the definition of $\mathcal{A}$.

In this proof, we say that a set B is "bad" if $|B \cap D| > 0.15n$. The number of permutations such that $\pi(S) = D$ is $(|S|)!(N - |S|)! = (n-c)!(9n+c)!$. As we assumed that $\text{pr} < 3/4$, the number of permutations π such that $\pi(S) = D$ and $\mathcal{A}(P_\pi, c')$ is "bad" is at least

$$0.25(n-c)!(9n+c)! \geq (n-c)!\sqrt{n}\left(\tfrac{9n+c}{e}\right)^{9n+c}. \tag{1}$$

We will prove that, by the properties of $\mathcal{A}$, the number of such permutations is much smaller achieving a contradiction to our assumption that $\text{pr} < 3/4$.

We first upper bound, for a given "bad" set B, the number of permutations π such that $\pi(S) = D$ and $\mathcal{A}(P_\pi, c') = B$. Notice that the output of the deterministic algorithm $\mathcal{A}(P_\pi, c')$ must contain all points in $\{p_{\pi(i)} : n+1 \leq i \leq 10n\}$ (otherwise the radius of the approximated solution is at least $2 \cdot \rho(N) \cdot d$, compared to at most d when taking all points in $\{p_{\pi(i)} : n+1 \leq i \leq 10n\}$ and additional c points). Thus, if a permutation π satisfies $\pi(S) = D$ and $\mathcal{A}(P_\pi, c') = B$, then $[N] \setminus B \subset D \cup \pi(I)$, which implies $[N] \setminus (B \cup D) \subset \pi(I)$. Letting $b \stackrel{\text{def}}{=} |B \cap D| \geq 0.15n$,

$$|\,[N] \setminus (B \cup D)\,| \;=\; N - |B| - |D| + |B \cap D| \;=\; 10n - (9n+c) - (n-c) + b \;=\; b.$$

Every permutation π satisfying $\pi(S) = D$ and $\mathcal{A}(P_\pi, c') = B$ has a fixed set of size b contained in $\pi(I)$, thus, the number of such permutations is at most

$$(|S|)!\binom{|I|}{b} b!(N - |S| - b)! = (n-c)!\binom{c}{b} b!(9n + c - b)!.$$

Taking $b = 0.15n$ can only increase this expression (as we require that a smaller set is contained in $\pi(I)$). Thus, noting that $c \leq n$, the number of permutations such that $\pi(S) = D$ and $\mathcal{A}(P_\pi, c') = B$ is at most $(n-c)!\binom{n}{0.15n}(0.15n)!(8.85n + c)!$. First, $\binom{n}{0.15n} \leq 2^{H(0.15)n} \leq (16)^{0.15n}$, where $H(0.15) \leq 0.61$ is the Shannon entropy. Thus, using Stirling approximation, the number of such permutations is at most

$$O\left(\sqrt{n}(0.3)^{0.15n}\right) \cdot \left((n-c)!\sqrt{n}\left(\frac{9n+c}{e}\right)^{9n+c}\right). \tag{2}$$

By Observation 2, all instances (P_π, c') for permutations π such that $\pi(S) = D$ are equivalent according to $\mathcal{R}_{k\text{-center-I}}$. Thus, since $\mathcal{A}$ leaks at most $0.015N$ bits, there are at most $2^{0.015N}$ possible answers of $\mathcal{A}$ on these instances, in particular, there are at most $2^{0.015N} = 2^{0.15n}$ "bad" answers. Thus, by (2), the number of permutations such that $\pi(S) = D$ and $\mathcal{A}(P_\pi, c')$ is a "bad" set is at most

$$O\left(2^{0.15n}\sqrt{n}(0.3)^{0.15n}\right) \cdot \left((n-c)!\sqrt{n}\left(\frac{9n+c}{e}\right)^{9n+c}\right) \qquad (3)$$

As the number of permutations in (3) is smaller than the number of permutations in (1), we conclude that $\mathrm{pr} \geq 3/4$. □

Combining Lemma 3 and Lemma 4, if $\mathcal{A}$ is a $\rho(N)$-approximation algorithm for k-center-I that leaks $0.015N$ bits, then Algorithm Close to Unique k-Center returns a set that is 0.7-close to the optimal solution with probability at least $3/4$, and by Theorem 3, this is impossible unless RP = NP.

In the full version of the paper we show that Algorithm Close to Unique k-Center finds a set close to the optimal solution even when $\mathcal{A}$ is randomize.

Theorem 5. *Let $\rho(n) \leq 2^{\mathrm{poly}(n)}$. If* RP $\neq$ NP, *every efficient $\rho(n)$-approximation algorithm for k-*center-I *(in the general metric version) must leak $\Omega(n)$ bits.*

4 Privacy of Clustering with respect to the Definition of [13]

Trying to get around the impossibility results, we examine a generalization of a definition by Indyk and Woodruff [13], originally presented in the context of near neighbor search. In the modified definition, the approximation algorithm is allowed to leak the set of approximated solutions to an instance. More formally, we use Definition 1, and set the equivalence relation $\mathcal{R}^\eta$ to include η-approximate solutions as well:

Definition 10. *Let L be a minimization problem with cost function* cost*. A solution w is an η-approximation for x if $\mathrm{cost}_x(w) \leq \eta \cdot \min_{w'}(\mathrm{cost}_x(w'))$. Let $\mathrm{appx}(x) \stackrel{\mathrm{def}}{=} \{w : w$ is an η-approximation for $x\}$. Define the equivalence relation $\mathcal{R}_L^\eta$ as follows: $x \equiv_{\mathcal{R}_L^\eta} y$ iff $\mathrm{appx}(x) = \mathrm{appx}(y)$.*

Note that Definition 10 results in a range of equivalence relations, parameterized by η. When $\eta = 1$ we get the same equivalence relation as before.

We consider the *coordinate* version of k-center. In the full version of this paper we show a threshold at $\eta = 2$ for k-center-C: (1) When $\eta \geq 2$, every approximation algorithm is private with respect to $\mathcal{R}^\eta_{k\text{-center-C}}$. (2) For $\eta < 2$ the problem is as hard as when $\eta = 1$.

5 Infeasibility of Approximation of Vertex Cover that Leaks Information

In [1], it was proven that if RP $\neq$ NP, then for every constant $\epsilon > 0$, every algorithm that $n^{1-\epsilon}$ approximates vertex cover must leak $\Omega(\log n)$ bits. In this paper we strengthen this result showing that if RP $\neq$ NP, then every algorithm that $n^{1-\epsilon}$-approximates vertex cover must leak $\Omega(n^\epsilon)$ bits. We note that this results is nearly tight: In [1], an algorithms that $n^{1-\epsilon}$-approximates vertex cover and leaks $2n^\epsilon$ bits is described. We will describe the infeasibility result in stages. We will start by describing a new proof of the infeasibility of deterministic private approximation of vertex cover, then we will describe the infeasibility of deterministic $n^{1-\epsilon}$-approximation of vertex cover that leaks at most αn^ϵ bits (where $\alpha < 1$ is a specific constant). In the full version of the paper we show the same infeasibility result for randomized algorithms.

5.1 Infeasibility of Deterministic Private Approximation of Vertex Cover

We assume the existence of a deterministic private approximation algorithm for vertex-cover and show that such algorithm implies that RP = NP. The idea of the proof is to start with an instance G of unique-vertex-cover and construct a new graph G_π. First, polynomially many isolated vertices are added to the graph. This means that any approximation algorithm must return a small fraction of the vertices of the graph. Next, the names of the vertices in the graph are randomly permuted. The resulting graph is G_π. Consider two permutations that agree on the mapping of the vertices of the unique-vertex-cover. The two resulting graphs are equivalent and the private algorithm must return the same answer when executed on the two graphs. However, with high probability on the choice of the renaming of the vertices, this answer will contain the (renamed) vertices that consisted the minimum vertex cover in G, some isolated vertices, and no other non-isolated vertices. Thus, given the answer of the private algorithm, we take the non-isolated vertices and these vertices are the unique minimum vertex cover. As unique-vertex-cover is NP-hard [21], we conclude that no deterministic private approximation algorithm for vertex exists (unless RP = NP).

The structure of this proof is similar to the proof of infeasibility of k-center-I, presented in Section 3.2. There are two main differences implied by the characteristics of the problems. First, the size of the set returned by an approximation algorithm for vertex-cover is bigger than the size of the minimum vertex cover as opposed to k-center where the approximation algorithm always returns a set of c centers (whose objective function can be sub-optimal). This results in somewhat different combinatorial arguments used in the proof. Second, it turns out that the roll of the vertices in the unique vertex cover of the graph is similar to the roll of the points *not* in the optimal solution of k-center. For example, we construct a new graph by adding isolated vertices which are not in the minimum vertex cover of the new graph.

We next formally define the construction of adding vertices and permuting the names. Given a graph $G = (V, E)$, where $|V| = n$, an integer $N > n$, and an injection $\pi : V \to [N]$ (that is, $\pi(u) \neq \pi(v)$ for every $u \neq v$), we construct a graph $G_\pi = ([N], E_\pi)$, where $E_\pi = \{(\pi(u), \pi(v)) : (u, v) \in E\}$. That is, the graph G_π is constructed by adding $N - n$ isolated vertices to G and choosing random names for the original n vertices. Throughout this section, the number of vertices in G is denoted by n, and the number of vertices in G_π is denoted by N. We execute the approximation algorithm on G_π, hence its approximation ratio and its leakage are functions of N. Notice that if G has a unique vertex cover C, then G_π has a unique vertex cover $\pi(C) \stackrel{\text{def}}{=} \{\pi(u) : u \in C\}$. In particular,

Observation 3. *Let G be a graph with a unique minimum vertex cover C, where $k \stackrel{\text{def}}{=} |C|$, and $\pi_1, \pi_2 : V \to [N]$ be two injections such that $\pi_1(C) = \pi_2(C)$. Then, $(G_{\pi_1}, k) \equiv_{\mathcal{R}_{\text{VC}}} (G_{\pi_2}, k)$.*

In Fig. 3, we describe an algorithm that uses this observation to find the unique minimum vertex cover assuming the existence of a private approximation algorithm for vertex cover. In the next lemma, we prove that Algorithm VERTEX COVER solves the unique-vertex-cover problem.

Algorithm VERTEX COVER:
Input: A Graph $G = (V, E)$ and an integer t.
Promise: G has a unique vertex cover of size t.
Output: The unique vertex cover of G of size t.

1. Let $N \leftarrow (4n)^{2/\epsilon}$.
2. Choose an injection $\pi : V \to [N]$ uniformly at random and construct the graph G_π.
3. Let $B \leftarrow \mathcal{A}(G_\pi)$ and $B^{-1} \leftarrow \{u \in V : \pi(u) \in B\}$.
4. Return B^{-1}.

Fig. 3. An algorithm that finds the unique minimum vertex cover.

Lemma 5. *Let $\epsilon > 0$ be a constant. If $\mathcal{A}$ is a deterministic $N^{1-\epsilon}$-private approximation algorithm for vertex cover and G has a unique vertex cover of size t, then, with probability at least $3/4$, Algorithm* VERTEX COVER *returns the unique vertex cover of G of size t.*

Proof. First, observe that B^{-1} is a vertex cover of G: For every $(u, v) \in E$ the edge $(\pi(u), \pi(v))$ is in E_π, thus at least one of $\pi(u), \pi(v)$ is in B and at least one of u, v is in B^{-1}. Notice that if $\pi(v) \notin \mathcal{A}(G_\pi)$ for every $v \in V \setminus C$, then Algorithm VERTEX COVER returns the vertex cover C. We will show that the probability of this event is at least $3/4$.

We say that an injection $\pi : V \to [N]$ *avoids* a set B if $\pi(v) \notin B$ for every $v \in V \setminus C$. See Fig. 4. By Obseration 3, the output B of the deterministic algorithm $\mathcal{A}$ depends only on $\pi(C)$. Thus, it suffices to show that for every possible value of D, the probability that a random injection π such that $\pi(C) = D$ avoids $B = \mathcal{A}(G_\pi)$ is at least $3/4$. As G_π has a cover of size at most n, and $\mathcal{A}$ is an

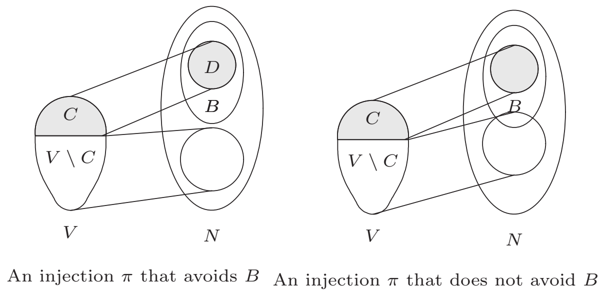

Fig. 4. Injections that avoid and do not avoid the output of $\mathcal{A}$.

$N^{1-\epsilon}$-approximation algorithm, $|B| \leq nN^{1-\epsilon}$. Thus, since $N = (4n)^{2/\epsilon}$,

$$\Pr[\pi \text{ avoids } B|\pi(C) = D] \geq \prod_{i=1}^{|V|-|C|} \left(1 - \frac{|B|}{N-n}\right) \geq \left(1 - \frac{nN^{1-\epsilon}}{N/2}\right)^n$$
$$= \left(1 - \frac{2n}{N^\epsilon}\right)^n = \left(1 - \frac{1}{8n}\right)^n > \frac{3}{4}.$$

To conclude, the probability that the random π avoids $\mathcal{A}(G_\pi)$ is at least 3/4. In this case $B^{-1} = C$ (as B^{-1} is a vertex cover of G that does not contain any vertices in $V \setminus C$) and the algorithm succeeds. □

Infeasibility of leaking $O(\log n)$ bits. Now, assume that Algorithm $\mathcal{A}$ is a deterministic $N^{1-\epsilon}$-approximation algorithm that leaks at most $(\epsilon \log N)/2$ bits. In this case, for every equivalence class of $\equiv_{\mathcal{R}_{\mathrm{VC}}}$, there are at most $2^{(\epsilon \log N)/2)} = N^{\epsilon/2}$ possible answers. In particular, for every possible value of D, there are at most $N^{\epsilon/2}$ answers for all graphs G_π such that the injection π satisfies $\pi(C) = D$. If the injection π avoids the union of these answers, then Algorithm Vertex Cover succeeds for a graph G that has a unique vertex cover of size t. The size of the union of the answers is at most $N^{\epsilon/2} \cdot nN^{1-\epsilon} = nN^{1-\epsilon/2}$, and if we take $N = (4n)^{4/\epsilon}$ in Algorithm Vertex Cover, then with probability at least 3/4 the algorithm succeeds for a graph G that has a unique vertex cover of size t. However, we want to go beyond this leakage.

5.2 Infeasibility of Approximation of Vertex Cover that Leaks Many Bits

Our goal is to prove that there exists a constant α such that for every constant $\epsilon > 0$, if RP $\neq$ NP, then there is no efficient algorithm that $N^{1-\epsilon}$-approximates the vertex cover problem while leaking at most $\alpha N^{1-\epsilon}$ bits. This is done by using the results of [16,5] that shows that it is NP-hard to produce a set that is close to a minimal vertex cover as defined in Section 2.5. Using this result, we only need that B^{-1} is close to the minimum vertex cover. We show that, even if $\mathcal{A}$ leaks many bits, for a random injection, the set B^{-1} is close to the minimum vertex cover.

> Algorithm Close to Vertex Cover:
> Input: A Graph $G = (V, E)$ and an integer t.
> Promise: G has a unique vertex cover of size t.
> Output: A set S that is δ-close to the unique vertex cover of G of size t for some constant $\delta > 1/2$.
>
> 1. Let $N \leftarrow (100n)^{1/\epsilon}$.
> 2. Choose a random injection $\pi : V \rightarrow [N]$ with uniform distribution and construct the graph G_π.
> 3. Let $B \leftarrow \mathcal{A}(G_\pi)$ and $B^{-1} \leftarrow \{u \in V : \pi(u) \in B\}$.
> 4. Return B^{-1}.

Fig. 5. An algorithm that returns a set close to a unique minimum vertex cover.

In Fig. 5, we describe Algorithm Close to Unique k-Center that finds a set close to the unique vertex cover of G assuming the existence of a deterministic $N^{1-\epsilon}$-approximation algorithm for vertex cover that leaks αN^{ϵ} bits. (In the full version of the paper we show how to generalize the analysis to deal with a randomized $N^{1-\epsilon}$-approximation algorithm.) To prove the correctness of the algorithm we need the following definition and lemma.

Definition 11. *Let $C \subset V$ be the unique minimum vertex cover of a graph G, and $\pi : V \rightarrow [N]$ be an injection. We say that π δ-avoids a set B if $|\{v \in V \setminus C : \pi(v) \in B\}| \leq \delta|V|$.*

Lemma 6. *Let $\epsilon > 0$ be a constant, and $B \subset [N]$, $D \subset [N]$ be sets, where $|B| \leq nN^{1-\epsilon}$. If $N = (100n)^{1/\epsilon}$ and an injection $\pi : V \rightarrow [N]$ is chosen at random with uniform distribution, then $\Pr[\pi$ does not 0.2-avoid $B|\pi(C) = D] \leq e^{-0.2n}$.*

The lemma is proved by using the Chernoff bound noting that the events $\pi(u) \in B$ and $\pi(v) \in B$ are "nearly" independent for $u \neq v$.

Lemma 7. *There exists a constant $\alpha < 1$ such that, for every constant $\epsilon > 0$, if $\mathcal{A}$ is a deterministic $N^{1-\epsilon}$-approximation algorithm for vertex cover that leaks at most αN^{ϵ} bits, then for every G and t such that G has a unique vertex cover of size t, with probability at least $3/4$, Algorithm* Close to Vertex Cover *returns a set that is 0.6-close to the minimum vertex cover of G.*

Proof (sketch): Let G and t be such that G has a unique vertex cover of size t; denote this vertex cover by C. We fix a set D and consider only injections π such that $\pi(C) = D$. Let $\alpha = 0.002$ and assume that $\mathcal{A}$ leaks at most $\alpha N^{\epsilon} = 0.2n$ bits (since $N = (100n)^{1/\epsilon}$). By Observation 3, if we restrict ourself to such injections, then the output of $\mathcal{A}$ has at $2^{0.2n}$ options. Denote these answers by $B_1, \ldots, B_\ell$ for $\ell \leq 2^{0.2n}$. By Lemma 6, for every possible value of B, the probability that a random injection π such that $\pi(C) = D$ does not 0.2-avoid B is at most $e^{-0.2n}$. Thus, by the union bound, the probability that a random injection π such that $\pi(C) = D$ 0.2-avoids $\mathcal{A}(G_\pi)$ is at least $1 - (2/e)^{0.2n} \gg 3/4$. In this case B^{-1} contains at most $0.2n$ vertices not from the minimum vertex cover C. Recall that B^{-1} is a vertex cover of G. Therefore, $|C \setminus B^{-1}| \leq 0.2n$ (as $|B^{-1}| > |C|$ and $|B^{-1} \setminus C| \leq 0.2n$). We conclude that B^{-1} is 0.6-close to a vertex cover of G as claimed. □

Theorem 6. *There exists a constant $\alpha > 0$ such that, if* RP $\neq$ NP, *there is no efficient $N^{1-\epsilon}$-approximation algorithm for vertex cover that leaks αN^ϵ bits.*

6 Discussion

The generic nature of our techniques suggests that, even if the notion of private approximations would be found useful for some NP-complete problems, it would be infeasible for many other problems. Hence, there is a need for alternative formulations of private approximations for search problems.

The definitional framework of [1] allows for such formulations, by choosing the appropriate equivalence relation on input instances. Considering vertex-cover for concreteness, the choice in [1] and the current work was to protect against distinguishing between inputs with the same set of vertex covers. A different choice, that could have been made, is to protect against distinguishing between inputs that have the same lexicographically first maximal matching. (In fact, the latter is feasible and allows a factor 2 approximation).

A different incomparable notion of privacy was pursued in recent work on private data analysis. For example, [4] present a variant on the k-means clustering algorithm that is applied to a database, where each row contains a point corresponding to an individual's information. This algorithm satisfies a privacy definition devised to protect individual information.

Finally, a note about leakage of information as discussed in this work. It is clear that introduction of leakage may be problematic in many applications (to say the least). In particular, leakage is problematic when composing protocols. However, faced by the impossibility results, it is important to understand whether a well defined small amount of leakage can help. For some functionalities allowing a small amount of leakage bypasses an impossibility result – approximating the size of the vertex cover [10], and finding an assignment that satisfies $7/8 - \epsilon$ of the clauses for exact max 3SAT [1]. Unfortunately, this is not the case for the problems discussed in this work.

Acknowledgments. We thank Enav Weinreb and Yuval Ishai for interesting discussions on this subjects and we thank the TCC program committee for their helpful comments.

References

1. A. Beimel, P. Carmi, K. Nissim, and E. Weinreb. Private approximation of search problems. In *Proc. of the 38th STOC*, pages 119–128, 2006.
2. M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computations. In *Proc. of the 20th STOC* pages 1–10, 1988.
3. L. Berman and J. Hartmanis. On isomorphisms and density of NP and other complete sets. *SICOMP*, 6:305–322, 1977.
4. A. Blum, C. Dwork, F. McSherry, and K. Nissim. Practical privacy: the SuLQ framework. In *Proc. of the 24th PODS*, pages 128–138, 2005.
5. U. Feige, M. Langberg, and K. Nissim. On the hardness of approximating NP witnesses. In *3rd APPROX*, volume 1913 of *LNCS*, pages 120–131. 2000.
6. J. Feigenbaum, Y. Ishai, T. Malkin, K. Nissim, M. J. Strauss, and R. N. Wright. Secure multiparty computation of approximations. *TALG*, 2(3):435–472, 2006.
7. M. J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 1–19. 2004.
8. O. Goldreich, S. Micali, and A. Wigderson. How to play any mental game. In *Proc. of the 19th STOC*, pages 218–229, 1987.
9. T. F. Gonzalez. Clustering to minimize the maximum inter-cluster distance. *TCS*, 38:293–306, 1985.
10. S. Halevi, R. Krauthgamer, E. Kushilevitz, and K. Nissim. Private approximation of NP-hard functions. In *Proc. of the 33th STOC*, pages 550–559, 2001.
11. D. S. Hochbaum and D. B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *JACM*, 533-550:33, 1986.
12. W. L. Hsu and G. L. Nemhauser. Easy and hard bottleneck location problems. *DAM*, 1:209–216, 1979.
13. P. Indyk and D. Woodruff. Polylogarithmic private approximations and efficient matching. In *TCC 2006*, volume 3876 of *LNCS*, pages 245–264. 2006.
14. O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems, part I: the p-centers. *SIAM J. Appl. Math*, 37:513–538, 1979.
15. E. Kiltz, G. Leander, and J. Malone-Lee. Secure computation of the mean and related statistics. In , *TCC 2005*, volume 3378 of *LNCS*, pages 283–302. 2005.
16. R. Kumar and D. Sivakumar. Proofs, codes, and polynomial-time reducibilities. In *Proc. of the 14th CCC*, pages 46–53, 1999.
17. M. Mitzenmacher and E. Upfal. *Probability and Computing.* Cambridge University Press, 2005.
18. J. Plesnik. On the computational complexity of centers locating in a graph. *Aplikace Matematiky*, 25:445–452, 1980.
19. J. Simon. On the difference between one and many. In *Proc. of the 4th ICALP*, volume 52 of *LNCS*, pages 480–491. 1977.
20. L. G. Valiant. A reduction from satisfiability to Hamiltonian circuits that preserves the number of solutions. Manuscript, Leeds, 1974.
21. L. G. Valiant and V. V. Vazirani. NP is as easy as detecting unique solutions. *TCS*, 47:85–93, 1986.
22. A. C. Yao. Protocols for secure computations. In *Proc. of the 23th FOCS*, pages 160–164, 1982.