The Power of the Dinur-Nissim Algorithm: Breaking Privacy of Statistical and Graph Databases

Krzysztof Choromanski
Department of Industrial Engineering and
Operations Research,
Columbia University
New York, USA
kmc2178@columbia.edu

Tal Malkin
Department of Computer Science,
Columbia University
New York, USA
tal@cs.columbia.edu

ABSTRACT

A few years ago, Dinur and Nissim (PODS, 2003) proposed an algorithm for breaking database privacy when statistical queries are answered with a perturbation error of magnitude $o(\sqrt{n})$ for a database of size n. This negative result is very strong in the sense that it completely reconstructs $\Omega(n)$ data bits with an algorithm that is simple, uses random queries, and does not put any restriction on the perturbation other than its magnitude. Their algorithm works for a model where the database consists of *bits*, and the statistical queries asked by the adversary are *sum queries* for a subset of locations.

In this paper we extend the attack to work for much more general settings in terms of the type of statistical query allowed, the database domain, and the general tradeoff between perturbation and privacy. Specifically, we prove:

- For queries of the type $\sum_{i=1}^{n} \phi_i x_i$ where ϕ_i are i.i.d. and with a finite third moment and positive variance (this includes as a special case the sum queries of Dinur-Nissim and several subsequent extensions), we prove that the quadratic relation between the perturbation and what the adversary can reconstruct holds even for smaller perturbations, and even for a larger data domain. If ϕ_i is Gaussian, Poissonian, or bounded and of positive variance, this holds for arbitrary data domains and perturbation; for other ϕ_i this holds as long as the domain is not too large and the perturbation is not too small.
- A positive result showing that for a sum query the negative result mentioned above is tight. Specifically, we build a distribution on bit databases and an answering algorithm such that any adversary who wants to recover a little more than the negative result above allows, will not succeed except with negligible probability.
- We consider a richer class of summation queries, focusing on databases representing graphs, where each entry is an edge, and the query is a structural function of a subgraph. We show an attack that recovers a big portion of the graph edges, as long as the graph and the function satisfy certain properties.

The attacking algorithms in both our negative results are straight-forward extensions of the Dinur-Nissim attack, based on asking ϕ-weighted queries or queries choosing a subgraph uniformly at random. The novelty of our work is in the analysis, showing that this simple attack is much more powerful than was previously known, as well as pointing to possible limits of this approach and putting forth new application domains such as graph problems (which may occur in social networks, Internet graphs, etc). These results may find applications not only for breaking privacy, but also in the positive direction, for recovering complicated structure information using inaccurate estimates about its substructures.

Categories and Subject Descriptors

E.4 [**Coding and Information Theory**]: Error Control Codes; H.2.8 [**Database Applications**]: Statistical Databases; G.3 [**Probability and Statistics**]: Probabilistic Algorithms

General Terms

Algorithms, Theory

Keywords

Data Privacy, Graph Privacy, Statistical Databases, Statistical Attacks, Blatant non-privacy

1. INTRODUCTION

Private data analysis aims to provide statistical information about the database while maintaining privacy of the records. Motivating applications abound, including settings where each database record contains medical, financial, or other private information of an individual. An emerging application domain that has received far less attention, is one where the database corresponds to a graph (say a social, wireless, or wired network graph) that one would like to compute statistics on, while maintaining some privacy of the topology. Typical examples of statistical information that may be provided include approximate sums (e.g., for counting or averaging), distribution parameters, histograms, etc.

PODS'12, May 21–23, 2012, Scottsdale, Arizona, USA.

A seminal paper by Dinur and Nissim [DN03] initiated a theoretically sound study of reasonable notions supporting both privacy and utility, followed by a large body of work (cf., [DMNS06, Dwo06, DKM$^+$06a, Dwo07, Dwo09, Dwo10, DNPR10, Yek10, DNP$^+$10, NST10, BN10, MM10, GLM$^+$10, GLM$^+$09, MM09, HT10]) addressing definitions, possibility, and impossibility results in different settings. The typical model (and the one that we use here), is one where the database is represented by a string $d = d_1, \dots, d_n$ held by a curator, who answers user queries using a mechanism A which first computes the correct answer, and then adds a perturbation according to some distribution (such a mechanism is indeed often used in practice).

The community has converged on the notion of differential privacy [DMNS06] as the right privacy notion for designing mechanisms balancing privacy and utility. However, in this paper we focus on attacks that violate privacy in a very strong way (beyond just violating differential privacy), following the approach introduced by [DN03] and described below. In this model, the attacker uses perturbated answers to queries on a database d to come up with another database x, such that with overwhelming probability, x is the same as the original d for a huge fraction of the entries. This is clearly not private for any reasonable notion of privacy, and is commonly referred to as "blatantly non-private". For the rest of the paper, this is what we will mean when we refer to breaking privacy (or "reconstructing all but a small fraction of entries").

The Dinur-Nissim Attack.

Dinur and Nissim [DN03] considered a database $d \in \{0,1\}^n$ consisting of *bits*, and *sum queries*, where the query is given in the form of a subset of indices, and the answer is the (perturbed) sum of data bits in those locations. They proved that if a mechanism A uses a perturbation whose magnitude is bounded by $o(\sqrt{n})$), then there is an efficient adversary that for every given $\sigma > 0$ can ask a polynomial number of queries and then reconstruct all but σn bits of the database with overwhelming probability. Specifically, the adversary's queries are simply uniformly chosen random subsets, which we will denote $\sum_{i=1}^{n} \phi_i d_i$ ("ϕ-weighted sums"), where each ϕ_i is chosen independently and uniformly from $\{0,1\}$. The output x is computed by solving the resulting linear program, and rounding each entry to $\{0,1\}$. We will refer to this as the *DN attack*. The proof relies on a "disqualifying lemma" that shows that this solution, with overwhelming probability, must agree with the original data d on all but an arbitrarily small linear fraction of entries. We discuss various extensions of the DN attack and other related work in Section 1.3.

1.1 Motivation and Goals

The main goal of our paper is to extend the DN attack to work for a wider variety of settings, focusing on the following aspects.

Queries: We want to extend the attack to other queries beyond the ones considered before, since in different application domains different statistical information about the data may be obtainable. It is arguably not reasonable to assume that the adversary has complete control over exactly which queries he may obtain answers to; in fact, in some cases the adversary may have no control whatsoever, and the information is just published by the curator using some distribution. The more flexibility there is regarding what type and distribution of queries are needed in order for the attack to succeed, the stronger the attack is.

Data domain: The DN attack and its extensions were analyzed for binary databases, where each entry is a bit. It is not hard to extend the attack to work for databases where each entry is taken from some constant size domain. We want to support larger domains (ideally, exponential size domains such as arbitrary polynomial-size strings, but it is not even clear how to extend the analysis of previous work for a polynomial size domain).

Tradeoff between perturbation and privacy compromise: The quadratic relation between the magnitude of the perturbation $\epsilon(n) = o(h(n))$ and what the attack can reconstruct (all but $O(h^2(n))$ entries) was proven in [DN03] for $h(n) = o(\sqrt{n})$. For a larger perturbation $\Omega(\sqrt{n})$ this relation is vacuously true, and in fact [DN03] show that in this case a reasonable notion of privacy can be achieved (as further developed in subsequent work to differential privacy). However, it is interesting to consider what happens if we allow *smaller* perturbations – can the attack be extended to reveal *even more than a linear fraction* of elements in the database? This question is especially relevant if one views this not as an attack, but as a way to recover data from noisy information, where a very small perturbation may be likely. This quadratic relation is indeed proved to hold for a specific type of queries in [DMT07] (see Section 1.3), but it is not clear how to extend their analysis (or the analysis of [DN03]) to handle random sum queries or other ϕ-weighted sum queries for smaller perturbations.

1.2 Our Results

We significantly extend the applicability of the DN attack along the fronts discussed above, in two settings. First, we consider more general ϕ-weighted sum queries, with a wider range of possible domains and perturbations. Second, we consider more complex summation queries, motivated by graph privacy applications. We prove both our attacks, as well as proving a positive result indicating the tightness of our first result. We provide more detail on each of these results below. The attacking algorithms in both our negative results are straight-forward extensions of the DN attack, asking simple randomized queries; we view this as a virtue of the attacks. The technical novelty of our work is in the analysis, showing that this simple attack is much more powerful than was previously known.

Our proof follows the same structure as [DN03], relying on a "disqualifying lemma" to prove that the rounded solution to the linear program agrees with the original database in many entries. However, [DN03] uses the Azuma inequality to prove their disqualifying lemma for random sums, while we have more general queries (including unbounded and non-i.i.d. summations), for which their techniques do not go through. Thus, we use (for different results) the Central Limit Theorem, utilizing a non-uniform version of the Berry-Esseen inequality [NT07] and a martingale version of Azuma's inequality.

We note that, although we present our results from the perspective of a privacy-breaking attack, they may also find applications in the positive direction, for recovering data from noisy information. This direction seems particularly promising in the context of structural databases such as graphs, where inaccurate information about substructures may be locally obtained, and used to deduce global information.

1.2.1 ϕ-weighted Sum Queries for Statistical Databases

A Negative Result. Consider the setting where each data element is taken from the domain $\{0, 1, \ldots, g(n)\}$ and queries are of the form $\sum_{i=1}^{n} \phi_i d_i$. Further consider any mechanism that adds a perturbation bounded by some $\epsilon(n) = o(h(n))$ for some $h(n)$. We prove that in this setting there is an efficient algorithm that can recover all but $O(h^2(n))$ entries, as long as either:

- ϕ_i are i.i.d. with a positive variance and a finite third moment, and $g(n) = o(n^{-\frac{1}{3}} h(n))$; or

- ϕ_i are i.i.d. random variables that are Gaussian, Poissonian, or bounded and with a positive variance.

Note that the second result holds for arbitrary size domains (even exponential) and a more limited class of ϕ, while the first result has more general ϕ, but holds only as long as the domain is not too large and the perturbation is not too small.[1]

We can use the first result to obtain separate generalizations of the DN attack to a larger domain or a smaller perturbation: Taking a binary $\{0, 1\}$ database gives the quadratic tradeoff as long as the perturbation satisfies $h(n) = \omega(n^{\frac{1}{3}})$. Taking $h(n) = \sqrt{n}$ gives the quadratic tradeoff as long as the domain satisfies $g(n) = o(n^{\frac{1}{6}})$. We can also use the first result to simultaneously improve both, recovering all but a sublinear number of entries, each taken from a sublinear size domain.

A Positive Result for Sums. We provide a positive result for sum queries showing that, roughly, for any given perturbation $o(h(n))$, the attacker cannot recover $n - o(\frac{h^2(n)}{\log n})$ bits, except with negligible probability. This matches the negative result above (recovering $n - O(h^2(n))$ bits) up to a logarithmic factor. Specifically, we build a distribution on bit databases and an answering mechanism for which we prove that no non-adaptive adversary can break the bound except with negligible probability. This holds for any non-adaptive adversary (even a computationally unbounded one) who asks polynomially many queries.

We emphasize that this positive result serves to show our negative result is tight, and not as a claim of privacy. Indeed, showing that blatant-non-privacy doesn't hold does not preclude other (possibly weaker) types of privacy violations.

Note that this result is in a different direction from the tightness result shown in [DN03]. They show that increasing the perturbation to $\Omega(\sqrt{n})$ will no longer let the attacker recover information as in their original attack. We show that for any perturbation in our range ($o(\sqrt{n})$ or lower) the adversary cannot recover any more than what our attack recovers.

1.2.2 Graph Databases

For the ϕ-weighted query setting considered so far, each element was selected according to some distribution ϕ_i independently of other elements. We now turn our attention to more complex queries, that may depend on structural properties and connections among the data elements. A particular motivation for us are graph databases, where the detailed topology of the graph should remain private, but some information about the graph may be released

We consider a setting where the query is a subset of indices, and the answer is (a perturbation of) some function h on the collection of all these entries together. In the most general form, our results can be presented as some conditions on this answer function h and the underlying data, such that an adaptation of the DN attack recovers most of the original data. However, we will present our results in a more narrow form, for reasons of readability and motivation (we mention several generalizations later in the paper). In particular, we will focus on databases representing graphs, and consider summation queries, where the entries included in the sum depend on the global structure of the graph. It is interesting to explore additional instantiations (for graphs and maybe also for other types of databases) where privacy is important and our conditions hold.

Let G_b (the "base graph") be an undirected public graph with m edges. We will consider a database of size m, where each entry is a (secret) weight of the corresponding edge. We will discuss only binary databases (weights of 0 or 1), but all our results can be readily extended for constant weights. We will denote by G_a the subgraph[2] of G_b obtained by considering only edges of weight 1. The graph G_a is the private information the adversary is trying to reconstruct. Intuitively, G_b represents what the adversary knows about the underlying graph (if he knows nothing, we can take G_b to be the complete graph), while which edge weights are non-zero (G_a) is what he is trying to find out.

The answer to a query consisting of a subset S is computed as follows. First, take the corresponding subgraph G_S and apply to it some (public) selection function Λ which selects which edges from G_S will participate in the sum. Importantly, Λ may select edges based on the structural properties of the graph (e.g., select only edges that are part of a 4-clique in the subgraph). Now, the (exact) answer is a function h which sums up, for all edges selected by Λ, some function f of their weights. That is, $h(G_S) = \sum f(w(e_i))$ where e_i are the edges selected by Λ and $w(e_i)$ are their weights. The mechanism then adds a perturbation bounded by $o(\sqrt{m})$ and outputs.

For this setting we prove that there is an efficient attack that asks uniformly random subgraph queries, and reconstruct a big portion of the data (weights), if the following conditions on h and the graph hold. (See later in the paper for more accurate statements).

- h is "not too sensitive": for any two data vectors v_1, v_2 that are close, $|h(v_1) - h(v_2)|$ is not too big. We have two such requirements for different notions of closeness, and an h that satisfies both of them is called *gradual*.

- h is "sensitive enough": recall that h is defined as the sum of a function f applied to the weight of each in-

[1] We do not know whether this lower bound on the size of the perturbation is inherent, or just a technical obstacle that can be overcome with a better analysis.

[2] All subgraphs in this paper are weak-subgraphs generated by the given edges, as opposed to (node-) induced subgraphs.

cluded edge. We require f to be *sensitive*, defined as having a slope that is bounded away from zero.

- Most edges in the graph are "not too sensitive" with respect to G, Λ, and h. This is a technical condition which very roughly says that, for most edges e, we want there to be a positive probability η that for a *random* subgraph G_S containing e, e is what we call (G_S, Λ)-active, which means: (1) Λ selects e to count towards the sum in h; and (2) removing e from G_S will not change which of the remaining edges are selected by Λ. Finally, this positive probability η should satisfy some equation in relation to sensitivity parameters of h.

Depending on the underlying structure of G_b and the functions h we are interested in, these conditions may or may not be satisfied, and may or may not be easy to check. We note that even when the conditions are hard to check, the attack can be applied, and the resulting reconstructed data has the guarantee that if the conditions happen to hold, then it agrees with the original graph database for a linear fraction of the entries. Our work is just a first step in this domain, and gives rise to several interesting open problems; in particular, identifying graph related problems where our conditions can be satisfied and that arise naturally (either in privacy-related settings, or in settings where we try to learn information about a graph).

To give a flavor of the scenarios where our attack can be mounted, consider the following example. Recall that an edge e in a graph is called a *bridge* if its removal increases the number of connected components in the graph. Consider a model where the answer to a query (subgraph) is the (perturbed) sum of weights for all the bridges in that subgraph. Let m be the length of the shortest cycle of the database graph G, let M be the length of the longest cycle in this graph, and assume that $m \geq 3$. Then as long as each edge in the graph is in no more than $\frac{2^{\frac{m}{2}-2}}{\sqrt{M}}$ cycles, all our conditions hold, and the adversary can reveal almost all weights of the edges in G. (See Section 3 for further details and other examples).

1.3 Related Work

The problem of releasing database statistics while preserving privacy has received much attention due to its fast-growing applicability and importance. Work on this topic dates back at least to the 80's (e.g., [DD82]), as well as more recent work such as [AS00, KMN05, NMK$^+$06, DKM$^+$06b, AFK$^+$10] and many others.

Not many models where graph queries are considered were analyzed so far, and to our knowledge, our paper is the first one providing an attack (blatant non-privacy) in this setting. Among privacy papers with "graph-database" model we should mention the very recent work of Gupta, Roth, and Ullman [GRU11], which addresses differential privacy of graph cuts. Roughly, each query can be identified with some subset of the vertices of the graph and an exact answer to that query is the cut induced by this subset; [GRU11] provide new algorithms solving the problem of approximately releasing the cut function of a graph while preserving differential privacy. It would be interesting to investigate how (and weather) our techniques can be applied to show a matching lower bound in the form of blatant non-privacy for this cut problem.[3]

There were several works extending the DN attack [DMNS06, DMT07, DY08, HT10, KRSU10], showing lower bounds on the perturbation necessary for certain databases, query functions and certain notions of privacy. Of most relevance for us is the attack of Dwork, McSherry and Talwar [DMT07], who show (among other things) that the quadratic relation between the perturbation and the number of entries that can be reconstructed holds in general for three types of attack queries of the form $\sum_{i=1}^{n} \phi_i d_i$ where ϕ_i are i.i.d.: either chosen uniformly at random from $\{-1, 0, 1\}$, from $\{-1, 1\}$, or chosen according to a Gaussian distribution with mean 0. (Recall that the DN attack can also be cast using these "ϕ-weighted sum" queries, where each ϕ_i is chosen uniformly at random from $\{0, 1\}$, corresponding to sums over random subsets). It is not clear how to extend the analysis of [DMT07] to apply to the setting of [DN03] or more general ϕ-hiding queries for smaller perturbations or polynomial size domains (which is one of our results).

Dwork and Yekhanin [DY08] use Fourier analysis to extend the DN attack to use fewer sum queries ($O(n)$ instead of $O(n \log^2(n))$) which are deterministic. Kasiviswanathan, Rudelson, Smith and Ullman [KRSU10] consider a database where each entry consists of several attributes (rather than a bit), and where each query is related to the so-called contingency table for a subset of attributes. They provide several types of lower bounds for this setting. The model in which a perturbated contingency table is released is very natural and has lots of applications. However, trying to compare to our results, it is not clear whether this model can be parameterized by some function ϕ fitting our conditions (i.e., finite third moment and positive variance) to obtain similar results for smaller perturbation error regimes.We also note that our approach uses much less advanced mathematical machinery than [KRSU10]. While [KRSU10] motivates models with queries of non-independent coefficients, they do not consider a graph database model and graph queries.

We note that while the original [DN03] attack is strictly a special case of ours, this is not so for follow up works such as [DMT07, DY08], who extend the DN attack on other fronts (e.g., allowing a small fraction of answers with unbounded perturbation, or optimizing the number of queries needed for the attack). On the other hand, our proofs do not require the heavy mathematical machinery utilized by [DMT07, DY08].

Finally, in a recent work (and independent of our own), Merener [Mer10] used similar mathematical tools (Berry-Esseen inequalities) to extend the DN attack. However, he focused on the analysis of the relation between the perturbation error added and the complexity of the adversary. He gave a formula for the number of queries used by the adversary as a function of the perturbation error. He considered databases with binary and real values from some bounded set. The database-access scheme he considered is similar to the one described in Dinur-Nissim paper. In contrast, we do not focus much on the complexity of the adversary. Our main goal is to prove that if the perturbation error is small enough the adversary can reveal much more than a linear

[3] We have not considered this yet, as we just found out about this work very recently. However, it seems that our theorems will not apply as-is, but that our techniques can potentially be used to directly attack this setting.

number of entries. Moreover, we consider database-access schemes using more complex functions defined on graphs.

2. RESULTS FOR ϕ-WEIGHTED QUERIES

In this section we describe our results for general ϕ-weighted sum queries, extended data domains and perturbations.

2.1 A Negative Result for Larger Domains and Smaller Perturbation Errors

Consider database vector d of length n, having entries from the set $\{0, 1, 2, ..., g(n)\}$. The database mechanism, given any query of the form $q = \{\phi_1, \phi_2, ..., \phi_n\}$, calculates $\sum_{i=1}^{n} \phi_i d_i$ and adds some perturbation error of magnitude $o(h(n))$ for some $h(n)$ (we often require $h(n)$ to be bounded from above by $O(\sqrt{n})$, because for a larger $h(n)$ the attack holds vacuously in a meaningless way, recovering all but n of the bits). Here ϕ_i's are independent copies of some random variable ϕ of positive variance and finite third moment. We give an efficient attack algorithm that allows to reveal all but $h^2(n)$ entries of a database with overwhelming probability $(1 - neg(n))$.

Definition 1. *We say that a random variable ϕ is primal if it is Gaussian, Poissonian or bounded and of positive variance.*

Theorem 1. *Let ϕ be a random variable of positive variance and finite third moment. If the database domain is of the form $\{0, 1, 2, ..., g(n)\}$, perturbation error $\epsilon = o(h(n))$ for some $h(n) = O(\sqrt{n})$ and $g(n) = o(n^{-\frac{1}{3}} h(n))$, then there is an efficient algorithm using $O(n \log^2(n))$ ϕ-weighted sum queries and revealing all but $h^2(n)$ entries of a database with probability (1-neg(n)). Moreover, if ϕ is primal the above holds for arbitrary $g(n)$.*

Plugging in $g(n) = 1$ for the first, and $h(n) = \sqrt{n}$ for the second, we obtain the following corollaries.

Corollary 1. *For a random variable ϕ of positive variance and finite third moment and a perturbation error $\epsilon = o(h(n))$ for $h(n) = O(\sqrt{n})$ such that: $h(n) = \omega(n^{\frac{1}{3}})$ there exists an efficient algorithm using ϕ-weighted sum queries and revealing all but $h^2(n)$ bits with probability (1-neg(n)), where neg(n) is some negligible function of n. Moreover, if ϕ is primal this holds for arbitrary $h(n)$.*

Corollary 2. *For a random variable ϕ of positive variance and finite third moment, any fixed $\sigma > 0$, perturbation error $\epsilon = o(\sqrt{n})$ and $g(n) = o(n^{\frac{1}{6}})$ there exists an efficient algorithm using ϕ-weighted sum queries and revealing all but σn entries with probability (1-neg(n)), where neg(n) is some negligible function of n. Moreover, if ϕ is primal this holds for arbitrary $g(n)$.*

Note that the original DN attack is a special case of both these corollaries.

The proof of Theorem 1 rests on the following "disqualifying lemma", which generalizes the [DN03] disqualifying lemma for sum functions.

Lemma 1 (Disqualifying Lemma). *Let ϕ be a random variable of finite third moment and positive variance. Let $x, d \in [0, 1]^n$ and $\epsilon = o(\frac{h(n)}{g(n)})$, where $g(n) = o(n^{-\frac{1}{3}} h(n))$. If $|\{i : |x_i - d_i| \geq \frac{1}{3g(n)}\}| > \frac{h^2(n)}{n}$, then $\exists \delta > 0$ such that for sufficiently large n we have:*

$$Pr_{q=\{\phi_1,\phi_2,...\phi_n\}}[|\sum_{i=1}^{n} \phi_i(x_i - d_i)| > 2\epsilon + 1] > \delta$$

Moreover, if ϕ is primal the inequality above holds for arbitrary $g(n)$ (without assuming $g(n) = o(n^{-\frac{1}{3}} h(n))$).

The proof appears in Appendix A.2. The high level idea is to prove that any vector x with many entries that are far from the corresponding entries in the database vector d, will with high probability be "disqualified" by one of the randomly chosen queries (namely, the answers on d and on x will be farther than the perturbation bound). Thus, any solution that "survives" all the queries and answers is with overwhelming probability very close to d on a large fraction of entries, and thus will be equal to d on those entries when rounded.

[DN03] proved their sum-function disqualifying lemma by using Azuma's inequality. For our ϕ-weighted queries where ϕ is bounded and of positive variance, we need to use a stronger version of Azuma's inequality to prove the lemma. However, for unbounded variables ϕ it is not clear how to use Azuma's inequality. In our proof we show that as long as we can prove that some sequence of random variables we define is uniformly-integrable then our proof goes through. The sequence is easily uniformly-integrable when ϕ is Gaussian or Poissonian, which completes the proof of those special cases. In the general case, we replace the use of Azuma's inequality with the use of the Central Limit Theorem, where the idea is to approximate the sum of random variables by a Gaussian. However, we need a good approximation, which we achieve by using a non-uniform version of the Berry-Esseen inequality [NT07] instead of the uniform one.

2.2 Positive results

Here we consider only bit databases and sum queries. From the last section we know that it is possible to extend the DN attack to smaller perturbation errors. A natural question is whether the parameters achieved can be improved, and in particular, whether for any fixed perturbation magnitude we may recover more bits than guaranteed by our proof. Dinur and Nissim [DN03] obtained some tightness result in their work, from a somewhat different direction. They considered perturbation error of order $o(\sqrt{n})$ (for which they reconstruct an arbitrarily large linear fraction of the database). They then show that if the perturbation order increases a little, it is no longer possible to reveal those data bits. On the other hand, we do not change the magnitude of the perturbation error. We consider perturbation error of magnitude $E = o(h(n))$ for some $h(n)$. If $h(n)$ is not too small then from Theorem 1 we know that there is an efficient adversary that reveals all but $O(h^2(n))$ bits. This adversary is *non-adaptive*, namely chooses the queries independently from the answers he already received. (In fact, the queries are simply random subsets of $\{1, 2, 3, ..., n\}$.) We ask whether, for the same perturbation magnitude, there is an attack algorithm that can reconstruct even more bits. Our next result shows that if the adversary is non-adaptive, he cannot reconstruct all but $o(\frac{h^2(n)}{\log(n)})$ bits (namely, our negative result was tight up to a logarithmic factor).

THEOREM 2. *Assume that perturbation error ϵ of a database algorithm is of the order $o(h(n))$, where $h(n) = O(\sqrt{n})$. Assume that the non-adaptive adversary chose polynomially many sum queries to ask. Then there is a probability distribution and an efficient database algorithm A such that the following holds: if database d was chosen from this distribution, $f(n) = o(\frac{h^2(n)}{\log(n)})$, algorithm A was used to add the perturbation error and $f(n) = \omega(1)$ then the adversary cannot guess all but $f(n)$ bits with probability greater than negligible.*

This result is especially interesting because it does not bound the adversary to run in a polynomial time. The only bound put on him is that he can ask only polynomially many queries. The proof uses some tricky distribution over databases. We prove that if we choose database using such a distribution then with probability almost 1 the adversary won't be able to reveal sufficiently many bits. To construct such a distribution we create a special combinatorial pattern – a family of subsets satisfying some conditions on their intersections. To prove that such a pattern exists we use the probabilistic method, namely randomly construct such a family and prove that with non-zero probability it satisfies all necessary conditions. The details of the construction can be found in the full version of the paper.

3. RESULTS FOR GRAPH FUNCTIONS

We now turn to consider more complex query functions. Although our results can also apply to other functions, we focus on databases representing graphs, as explained in Section 1.2.2.

3.1 Graph model and basic definitions

We consider here an undirected weighted graph G. The adversary works with a graph with edge weights from the discrete set $\{0, 1\}$ (this can be generalized to constant size domains). The query-function is defined for every weighted graph G with weights from the interval $[0, 1]$. We emphasize here that nonedges are not equivalent to edges with weight 0. We associate with a weighted graph G the "base graph" G_b which is the underlying unweighted graph, and the "active graph" G_a, which is the graph formed by edges of positive weight. G_b is public, while G_a is not (and this is what the adversary is trying to recover). We will think of the database as consisting of m entries, where m is the number of edges in G_b. Each entry will contain the 0/1 weight of the corresponding edge. A query is a subset of edges of G_b, and the exact answer is the value of some function h on the weighted graph created by this subset of edges. As usual, a perturbation is added to the output before it is released (here we will use a perturbation of magnitude $o(\sqrt{m})$)

We consider an output function h that is defined by summing up $\sum f(w_e)$ for *some* of the edges in the query subset, where w_e is the weight of the edge e (the value in the database), and f is some function (e.g., the identity, in which case we are just summing weights). To determine which edges e will participate in the sum, we use some selection function Λ that may depend on the structure of the graph. Our rigorous results and proof that our attack works efficiently, apply to what we call *the basic setting*, where Λ is only allowed to rely on the structure of query subgraph (which is public, as it is a subgraph of G_b), but is *not* allowed to depend on the weights. For example, Λ may be select edges that are bridges in the subgraph (namely edges whose removal will disconnect the subgraph). The *complex setting* is one where Λ is also allowed to depend on the structure of the hidden G_a. We discuss this setting later.

Notations. The set of all edges of a weighted graph G will be denoted by E. For a weighted graph G and an edge e we will denote by $G-e$ the graph obtained by deleting e from G. For a weighted edge e and a graph G we will denote by $G+e$ the graph obtained by adding edge e to G. In the analogous way we define operations: $G+E$ and $G-E$, where G is a graph and E is a set of edges. While deleting or adding an edge we always delete/add weight associated with this edge. The weight of an edge e will be denoted as w_e. For the set of edges $q \subseteq E$ we denote by $G_{|q}$ the subgraph of G obtained by considering only edges from q. The output of the function h on a graph G will be denoted as $h(G)$. For $E_s \subseteq E$ we denote by $h(E_s)$ the output of h on a graph determined by the set E_s. Let S be a subset of indices of a vector d representing some graph G. We denote by $h^d(S)$ the value of the function h on the subgraph of a graph represented by vector d, obtained by choosing edges of G related to the indices from S. Sometimes we will get rid of d and use shorter denotation: $h(S)$. Having some fixed graph G for which we enumerated all m edges we consider values of the function h on subgraphs of G. Every such subgraph will be denoted by a vector of length m where we put special symbol $\cdot$ for every index related to the edge that wasn't chosen to the subgraph. We need to extend arithmetical operations on real numbers to take into account also this new symbol. We assume that subtracting $\cdot$ from $\cdot$ gives 0. We extend linear order on real numbers such that $\cdot$ is less than every real number. Denote $R_e = R \bigcup \{\cdot\}$.

DEFINITION 2. *We say that 2 vectors v_1 and v_2 of length m are* similar *if they have special symbol $\cdot$ on the same entries. Intuitively, they represent subgraphs of the same structure but possibly different weights on edges.*

DEFINITION 3. *Function $h : R_e^m \rightarrow R$ is called* gradual *if it satisfies the following conditions:*

- $\exists_{w>0}|h(v_1) - h(v_2)| \leq w$ *for every two vectors v_1, v_2 that differ on at most one coordinate*
- *for every pair of two similar vectors: $a = (a_1, a_2, ..., a_m)$, $b = (b_1, b_2, ..., b_m)$ such that $\forall_{i\in\{1,2,...,m\}}|a_i - b_i| \leq \frac{1}{m}$ we have $|h(a) - h(b)| = o(\sqrt{m})$*

DEFINITION 4. *A function $f : R \rightarrow R^+ \bigcup\{0\}$ is sensitive if $\inf_{\Delta>0,x} \frac{f(x+\Delta)-f(x)}{\Delta} \geq c_f$ for some $c_f > 0$.*

DEFINITION 5. *An edge e of a graph G is (G,h,A,B)-gradual for $A, B > 0$ if for any subgraph H of the graph G-e we have:*

- $h(H + e) - h(H) \leq B$
- $h(H + e) - h(H) \geq -A$

Intuitively this definition says that function h is not too sensitive on the edge e. So the absence of an edge e does not change much the value of function h.

Definition 6. *We say that a graph is* labelled *if its vertices have labels (say, $1, \ldots, n$ for an n-vertex graph). We say two labelled graphs g_1, g_2 are* isomorphic *if they are isomorphic as unlabeled graphs, and there is an isomorphic embedding that maps vertices of g_1 to vertices of g_2 with the same labels (that is, if the isomorphic mapping maintains both node labels and edge structure).*

We say that a labelled graph g is isomorphic to some subgraph of labelled graph G if there is a (labelled) subgraph G_S of G such that g and G_S are isomorphic (as labelled graphs).

We now define when an edge e is (G_S, Λ)-active for a (public) selection function Λ in the general setting of labelled graphs (we may view unlabeled graphs as a special case).

Definition 7. *Take a labelled weighted graph G. Consider unweighted labelled graph G_b. Denote by G_b^S the set of all subgraphs of G_b. Consider the function $\Lambda : G_b^S \to E^S \to \{0, 1\}$ that for every subgraph G_S of G_b outputs a function that maps the set of edges of G_S into the binary set $\{0, 1\}$. We call Λ the selecting function because for every edge e of the subgraph G_S it determines whether e is selected (i.e $\Lambda(G_S)(e) = 1$) or not.*

Whenever $\Lambda(G_S)(e) = 1$ we say that e is (G_S, Λ)-acceptable. Denote by $A(G_S, \Lambda)$ the set of all (G_S, Λ)-acceptable edges. Whenever we have: $A(G_S, \Lambda) = A(G_S - \{e\}, \Lambda) + \{e\}$ for $e \in G_S$ we say that an edge e is (G_S, Λ)-active.

In other words, an edge e is (G_S, Λ)-active if it is selected by Λ (i.e., it is (G_S, Λ)-acceptable), and removing it from G_S will not change which other edges in G_S are selected by Λ.

3.2 Negative Results for Graph Models

We are ready to state our main result of this section (defined for the *basic setting* which we already explained informally above). The intuitive meaning of this theorem was given in the Introduction (Section 1.2.2).

Theorem 3. *Let G be some weighted undirected database graph of m edges with weights taken from the discrete set $\{0, 1\}$. Fix some public sensitive function f and selecting function Λ. Let h be a gradual function defined such that the output of h on a subgraph G_S is the sum of outputs of f on the weights of (G_S, Λ)-acceptable edges. Assume each edge of G is (G, h, A, B)-gradual for some positive constants A, B. Denote by P_e the conditional probability that e is not (G_S, Λ)-active in a random subgraph G_S of G given that e is in G_S. If for every edge e of a graph G we have: $P_e \leq 1 - \eta$ for some $\eta > 0$ such that $\frac{\eta}{1-\eta} > \frac{2(A+B)}{c_f}$ then for every fixed $\sigma > 0$ there exists a linear program of polynomial size that when solved enables the adversary to reveal all but σm weights of edges of a graph G with probability 1-neg(m) as long as the perturbation error added by database algorithm is of the order $o(\sqrt{m})$. Furthermore, this program can be constructed by the adversary by asking queries choosing uniformly at random subgraphs of a given graph.*

We discuss applications of the theorem below. We note that the theorem can be generalized, e.g., by relaxing the condition that all edges e have a positive probability to be active, to require this for almost all edges (the price is the quality of the attack – a slightly smaller, though still linear, fraction of the entries is revealed). This follows from the proof of Theorem 3, which is given in the full version of the paper.

The proof uses an attack similar to the DN attack in that it consists of polynomially many random subset queries, and then the resulting program is solved, and the solution rounded. We prove this by establishing a disqualifying lemma for a graph model, using a Martingale based version of Azuma's inequality. While that proof pertains to the basic setting, the mathematical machinery that we develop may be useful for the complex setting as well.

The challenges when moving to the complex setting are the following. First, once we ask random queries and obtain perturbated answers, the resulting program may not be solvable in polynomial time. Next, even if we manage to find a solution (or assume we have access to a solution oracle), it may be that the solution will in fact not let us reconstruct the original data. However, if a solution (in the complex setting) is found, and if it satisfies certain conditions, then it can be rounded and outputted; if it doesn't satisfy the conditions, we can iterate again, asking another set of random queries. If at any point we have a solution to the program, and the solution satisfies some conditions, then we can reconstruct most of the original database. This may be useful for applications where we can prove that a solution can be found, the condition can be checked, and a good solution will be arrived after a reasonable number of iterations. We leave it as an open problem to provide provable solutions for problems in the complex setting (general classes of such problems, or specific useful instantiations). For example, it would be interesting to find useful applications in the complex scenario where the program can be efficiently solved.

Applications.

Clearly, the sum function (as in the DN attack) is a special case where Λ always selects every edge (in particular, the edge selection does not rely on any structural properties). Of course, the power of this theorem is in allowing structure dependent queries.

Examples include scenarios in which an entry can contribute to the query result only when it is selected with edges creating a special configuration. Maybe it is the case that the publisher of database statistics does not want to give in the same time information of some special subset of entries that are crucial from his point of view so he creates for such a subset a forbidden pattern. It seems that it makes the goal of the adversary much more difficult. Our result gives sufficient conditions for a structure of the database graph that allows to completely break privacy. Moreover, under those conditions privacy is broken by solving a simple linear program that is completely analogous to the one used in the DN attack. Below we give two example application domains of our result.

Labelled-Graph model.

Consider a setting where the database answering mechanism stores a public finite set of labelled forbidden graphs F, each of at least one edge. For each forbidden graph $f \in F$ one of the edges of f is fixed, denote it by $e(f)$. The information which edge is fixed is also public. The function h on input G_S outputs the sum of weights of edges that satisfy the following property:

There is no isomorphic embedding that maps a forbidden graph f from F into some subgraph of G_S and $e(f)$ into this edge.

A perturbation error of order $o(\sqrt{m})$ is added (where m is the number of edges of G, namely the size of the database). Assume furthermore that each edge of the database graph is contained in at most A subgraphs of G that are isomorphic to some graph from F. We need one more condition for each edge e, namely:

$$\sum_{g \in F^e} \frac{1}{2^{e_g - 1}} \leq 1 - \eta, \tag{1}$$

where F^e is the set of all subgraphs of G, isomorphic to some graph from F and containing e, e_g is the number of edges of g and η is chosen such that $\frac{\eta}{1-\eta} > 2(A+1)$.

In such a scenario for every given $\sigma > 0$ there exists efficient algorithm that reveals all but σm edges (weights) of G with probability $1 - neg(m)$.

Bridge-counter model.

We define h as a function that for a subgraph G_S outputs the sum of weights of all its bridges. We may think about h as a sum function that takes into consideration only 'important edges' where important are edges that are bridges. As in the previous case, in order to break the privacy we need to assume few more things. For each edge e define by L_e the set of all edges e_1 such that e and e_1 are both edges of some cycle. We assume that: $\forall_e |L_e| \leq A$ for some fixed A.

The last condition that each edge of the graph should satisfy, analogous to the one from the previous example, is now of the form:

$$\sum_{c \in C_e} \frac{1}{2^{|c|-1}} \leq 1 - \eta, \tag{2}$$

where C_e is the set of all cycles of G containing e and once more η is satisfying: $\frac{\eta}{1-\eta} > 2(A+1)$.

For such a model, as before, for every given $\sigma > 0$ there exists efficient algorithm that reveals all but σm edges of G with probability 1-neg(m).

REMARK 1. *If we denote a girth of a hidden graph by g then the inequality 2 may be replaced by a stronger one, namely: $\sum_{c \in C_e} \frac{1}{2^{g-1}} \leq 1 - \eta$. So we require each edge e to be in no more than $2^{g-1}(1-\eta)$ cycles of G.*

REMARK 2. *Denote by g the girth of a hidden database graph and by M the length of the longest cycle. Assume that $g \geq 3$. Then the conditions above may be easily replaced by a stronger one, namely: each edge of G is in no more than $\frac{2^{\frac{g}{2}-2}}{\sqrt{M}}$ cycles of G. If this condition is satisfied, then the privacy of a database can be broken.*

Acknowledgments

We thank Sharon Goldberg, Moritz Hardt, Kobbi Nissim, and Adam Smith for several illuminating discussions and pointers.

4. REFERENCES

[AFK+10] Gagan Aggarwal, Tomas Feder, Krishnaram Kenthapadi, Samir Khuller, Rina Panigrahy, Dilys Thomas, and An Zhu. Achieving anonymity via clustering. *ACM Transactions on Algorithms (TALG)*, 6, 2010.

[AS00] Rakesh Agrawal and Ramakrishnan Srikant. Privacy-preserving data mining. 2000.

[BN10] Hai Brenner and Kobbi Nissim. Impossibility of differentially private universally optimal mechanisms. *CoRR*, abs/1008.0256, 2010.

[Cha10] Moses Charikar, editor. *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January 17-19, 2010*. SIAM, 2010.

[DD82] Dorothy E. Denning and Peter J. Denning. *Cryptography and Data Security*. 1982.

[DKM+06a] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *EUROCRYPT*, volume 4004 of *Lecture Notes in Computer Science*, pages 486–503. Springer, 2006.

[DKM+06b] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *Advances in Cryptology - EUROCRYPT 2006, 25th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 486–503, 2006.

[DMNS06] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *TCC*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006.

[DMT07] Cynthia Dwork, Frank McSherry, and Kunal Talwar. The price of privacy and the limits of LP decoding. In David S. Johnson and Uriel Feige, editors, *STOC*, pages 85–94. ACM, 2007.

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS)*, pages 202–210. ACM, 2003.

[DNP+10] Cynthia Dwork, Moni Naor, Toniann Pitassi, Guy N. Rothblum, and Sergey Yekhanin. Pan-private streaming algorithms. In Andrew Chi-Chih Yao, editor, *ICS*, pages 66–80. Tsinghua University Press, 2010.

[DNPR10] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In Schulman [Sch10], pages 715–724.

[Dwo06] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *ICALP (2)*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[Dwo07] Cynthia Dwork. Ask a better question, get a better answer a new approach to private data analysis. In Thomas Schwentick and Dan Suciu, editors, *ICDT*, volume 4353 of *Lecture Notes in Computer Science*, pages 18–27. Springer, 2007.

[Dwo09] Cynthia Dwork. The differential privacy frontier (extended abstract). In Omer Reingold, editor, *TCC*, volume 5444 of *Lecture Notes in Computer Science*, pages 496–502. Springer, 2009.

[Dwo10] Cynthia Dwork. Differential privacy in new settings. In Charikar [Cha10], pages 174–183.

[DY08] Cynthia Dwork and Sergey Yekhanin. New efficient attacks on statistical disclosure control mechanisms. In David Wagner, editor, *CRYPTO*, volume 5157 of *Lecture Notes in Computer Science*, pages 469–480. Springer, 2008.

[GLM+09] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private approximation algorithms. *CoRR*, abs/0903.4510, 2009.

[GLM+10] Anupam Gupta, Katrina Ligett, Frank McSherry, Aaron Roth, and Kunal Talwar. Differentially private combinatorial optimization. In Charikar [Cha10], pages 1106–1125.

[GRU11] Anupam Gupta, Aaron Roth, and Jonathan Ullman. Iterative constructions and private data release. *CoRR*, abs/1107.3731, 2011.

[HT10] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In Schulman [Sch10], pages 705–714.

[KMN05] Krishnaram Kenthapadi, Nina Mishra, and Kobbi Nissim. Simulatable auditing. *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS)*, 2005.

[KRSU10] Shiva Prasad Kasiviswanathan, Mark Rudelson, Adam Smith, and Jonathan Ullman. The price of privately releasing contingency tables and the spectra of random matrices with correlated rows. In Schulman [Sch10], pages 775–784.

[Mer10] Martin Merener. Polynomial-time attack on output perturbation sanitizers for real-valued datasets. *Journal of Privacy and Confidentiality*, 2(2):65–81, 2010.

[MM09] Frank McSherry and Ilya Mironov. Differentially private recommender systems: Building privacy into the netflix prize contenders. In John F. Elder IV, Françoise Fogelman-Soulié, Peter A. Flach, and Mohammed Javeed Zaki, editors, *KDD*, pages 627–636. ACM, 2009.

[MM10] Frank McSherry and Ratul Mahajan. Differentially-private network trace analysis. In Shivkumar Kalyanaraman, Venkata N. Padmanabhan, K. K. Ramakrishnan, Rajeev Shorey, and Geoffrey M. Voelker, editors, *SIGCOMM*, pages 123–134. ACM, 2010.

[NMK+06] Shubha U. Nabar, Bhaskara Marthi, Krishnaram Kenthapadi, Nina Mishra, and Rajeev Motwani. Towards robustness in query auditing. *VLDB '06 Proceedings of the 32nd international conference on Very large data bases*, 2006.

[NST10] Kobbi Nissim, Rann Smorodinsky, and Moshe Tennenholtz. Approximately optimal mechanism design via differential privacy. *CoRR*, abs/1004.2888, 2010.

[NT07] K. Neammanee and P. Thongtha. Improvement of the non-uniform version of berry-esseen inequality via paditz-siganov theorems. *Journal of Inequalities in Pure and Applied Mathematics (JIPM)*, 8(4), 2007.

[Sch10] Leonard J. Schulman, editor. *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*. ACM, 2010.

[She96] Ross M. Sheldon. *Stochastic Processes.* Wiley, 1996.

[Yek10] Sergey Yekhanin. Private information retrieval. *Commun. ACM*, 53(4):68–73, 2010.

APPENDIX

A. RESULTS FOR ϕ-WEIGHTED QUERIES: RIGOROUS EXPOSITION AND PROOFS

In this section we consider *ϕ-weighted sum queries*, where the query is defined by a random string $\{\phi_1, ..., \phi_n\}$, where ϕ_i's are independent copies of some random variable ϕ. The exact answer to the query is of the form: $\sum_{i=1}^{n} \phi_i d_i$ (which as usual will be perturbed by the mechanism). When ϕ takes values from the discrete set $\{v_1, ..., v_k\}$, each with the same probability $\frac{1}{k}$, we call the corresponding query a $\{v_1, ..., v_k\}$-query. The $\{0, 1\}$-query is a *sum query.* The Gaussian query is a ϕ-weighted sum query with ϕ being a Gaussian random variable. We assume that ϕ has finite third moment and positive variance.

For these queries, we will consider an extension on the DN attack, showing the quadratic relation holds even for small perturbations and all but sublinear entries revealed, as well as for larger domains, where each database entry is taken from the set $\{0, 1, 2, ..., g(n)\}$, where $g(n)$ is some function of n. The key part of the proof is a modified version of the so-called disqualifying lemma proposed by Dinur and Nissim in [DN03]. Before stating it we will describe the algorithm proposed by Dinur and Nissim and introduce some useful notation.

A.1 Database model and algorithms breaking privacy

Definition 8. *Let ϕ be a random variable of finite third moment and positive variance. Each query is of the form $\{\phi_1, ..., \phi_n\}$, where ϕ_i's are independent copies of ϕ and n is the size of a database. By a database $D(d, A, \epsilon, g(n))$ we mean vector d of n entries from a discrete set $\{0, 1, 2, ...g(n)\}$ together with the algorithm A, possibly randomized, that given any query $q = \{\phi_1, ..., \phi_n\}$ returns number $\hat{a}_q$ such that:*

$$|\hat{a}_q - \sum_{i=1}^{n} \phi_i d_i| \leq \epsilon$$

Fix some $\sigma > 0$. The aim of the adversary is to reveal all but σn entries of a database. In the following algorithm we assume that each entry of a database is either 0 or 1 so this is a special case of our model for g(n)=1. The following algorithm was proposed by Dinur and Nissim to achieve this goal for sum queries:

- Let $t = cn \log^2 n$ for some constant c that depends on chosen σ. For $1 \leq j \leq t$ choose uniformly at random $q_j \subseteq \{1, 2, ...n\}$ and get answer $\hat{a}_{q_j}$ from the database algorithm.
- Solve the following linear program with unknown $c_1, ..., c_n$:

$$\hat{a}_{q_j} - \epsilon \leq \sum_{i \in q_j} c_i \leq \hat{a}_{q_j} + \epsilon$$

$$0 \leq c_i \leq 1$$

for $1 \leq j \leq t$ and $1 \leq i \leq n$

- Let $x_i = 1$ if $c_i > \frac{1}{2}$ and $x_i = 0$ otherwise. Output vector x.

It turns out that as long as $\epsilon = o(\sqrt{n})$ the output vector x is exactly the same as d on all but σn bits with probability (1-neg(n)), where neg(n) is some negligible function of n.

Below we give a version of the algorithm for general ϕ-weighted sum queries (where ϕ satisfies the above conditions), a database with domain $\{0, 1, 2, ...g(n)\}$, where $g(n)$ is not necessarily 1, and perturbation error of the magnitude $o(h(n))$ for some $h(n) = O(\sqrt{n})$. We call this algorithm - the extended Dinur-Nissim algorithm:

- Let $t = cn \log^2 n$ for some big enough constant c appropriately chosen. For $1 \leq j \leq t$ take a query $q_j = \{\phi_1^j, \phi_2^j, ...\phi_n^j\}$ and get answer $\hat{a}_{q_j}$ from the database algorithm.
- Solve the following linear program with unknown $c_1, ..., c_n$:

$$\hat{a}_{q_j} - \epsilon \leq \sum_{i=1}^{n} \phi_i^j c_i \leq \hat{a}_{q_j} + \epsilon$$

$$0 \leq c_i \leq g(n)$$

for $1 \leq j \leq t$ and $1 \leq i \leq n$

- $\forall_i$ let x_i be obtained by rounding c_i to the nearest integer from $\{0, 1, 2, ...g(n)\}$. Output vector x.

The proof of the correctness of the first algorithm is based on the fact that for every vector that differs too much from d with some probability bounded from below by some constant greater than 0 the randomly chosen query will disqualify it. The precise analysis is contained in the so-called disqualifying lemma proposed and proved by Dinur and Nissim in the same paper. Later we will give our version of the disqualifying lemma that suites the situation when we want to reveal all but sublinear number of bits, we have larger domains for database entries and more general queries. As an immediate corollary we get:

Theorem 1. *Let ϕ be a random variable of positive variance and finite third moment. If the database domain is of the form $\{0, 1, 2, ..., g(n)\}$, perturbation error $\epsilon = o(h(n))$ for some $h(n) = O(\sqrt{n})$ and $g(n) = o(n^{-\frac{1}{3}} h(n))$, then there is an efficient algorithm using ϕ-weighted sum queries and revealing all but $h^2(n)$ entries of a database with probability (1-neg(n)). Moreover, if ϕ is primal the above holds for arbitrary $g(n)$.*

Plugging in $g(n) = 1$ gives Corollary 1, generalizing the DN attack on binary databases to work also for a smaller perturbation, as long as $h(n) = \omega(n^{\frac{1}{3}})$. Plugging in $h(n) = \sqrt{n}$ gives Corollary 2, generalizing the DN attack with $o(\sqrt{n})$ perturbation to work also for a larger domain, as long as $g(n) = o(n^{\frac{1}{6}})$.

The reason why Theorem 1 follows from our version of Disqualifying Lemma lies in the same analysis that was done by Dinur and Nissim in [DN03] so we skip it.

A.2 Disqualifying lemma for small perturbation errors

In this section we prove the so-called disqualifying lemma for the case when perturbation error for unscaled vectors is of the order $o(h(n))$ for some $h(n)$. We do not assume that $h(n) = \sqrt{n}$ as Dinur and Nissim did. Our result can be applied for h(n) of order $o(\sqrt{n})$ as long as h(n) is not of 'too small' order. In this case the lemma enables us to reveal much more than all but θn entries for some fixed small θ. We will use the notation introduced before. Especially, by ϵ we denote the perturbation error. By $g(n)$ we denote the maximal value that database entry can take.

Lemma 1 (Disqualifying Lemma). *Let ϕ be a random variable of finite third moment and positive variance. Let $x, d \in [0, 1]^n$ and $\epsilon = o(\frac{h(n)}{g(n)})$, where $g(n) = o(n^{-\frac{1}{3}} h(n))$. If $|\{i : |x_i - d_i| \geq \frac{1}{3g(n)}\}| > \frac{h^2(n)}{n}$, then $\exists \delta > 0$ such that for sufficiently large n we have:*

$$Pr_{q=\{\phi_1, \phi_2, ...\phi_n\}}[|\sum_{i=1}^{n} \phi_i (x_i - d_i)| > 2\epsilon + 1] > \delta$$

Moreover, if ϕ is primal the inequality above holds even if we do not assume that $g(n) = o(n^{-\frac{1}{3}} h(n))$.

Proof. We use the following denotation:

- Let $X_i = \phi_i (x_i - d_i)$. Random variables X_i for $i = 1, 2, ...n$ are independent because ϕ_i's are independent.
- Let $Z_i = X_i - E(X_i)$ for i=1,2,...n. So we have: $E(Z_i) = 0$ and $Var(Z_i) = Var(X_i)$
- Let $X = \sum_{i=1}^{n} X_i$
- Let $Y = (X - EX)^2$. In particular: $EY = Var(X) = \sum_{i=1}^{n} Var(X_i)$, because X_i are independent r.v

So to prove lemma we only need to prove that: $Pr[|X| > 2\epsilon + 1] > c$ for some positive constant c. We introduce constant T. Its exact numerical value will be determined later. Having it, we will consider two cases.

A.2.1 Case 1: $|E(X)| \leq T \sqrt{\sum_{i=1}^{n} Var(Z_i)}$

In this case we prove that:

$$Pr[|X - EX| > 2T \sqrt{\sum_{i=1}^{n} Var(Z_i)}] > C$$

for some positive constant C. This proves lemma because: $\sum_{i=1}^{n} Var(Z_i) \geq \sigma \frac{h^2(n)}{g^2(n)}$ for some $\sigma > 0$ and $\epsilon = o(\frac{h(n)}{g(n)})$.

The lower bound on $\sum_{i=1}^{n} Var(Z_i)$ is a direct implication of the fact that $Var(\phi) > 0$.

We partition the probability space into three regions:

- $A = \{Y < \alpha \sum_{i=1}^{n} Var(Z_i)\}$
- $B = \{\alpha \sum_{i=1}^{n} Var(Z_i) \leq Y \leq \beta \sum_{i=1}^{n} Var(Z_i)\}$
- $C = \{Y > \beta \sum_{i=1}^{n} Var(Z_i)\}$,

where constants $0 < \alpha < \beta$ will be determined later. Note that $E(Y) = \sum_{i=1}^{n} Var(Z_i)$.

We have:

$$E(Y) = Pr[A]E(Y|A) + Pr[B]E(Y|B) + Pr[C]E(Y|C) \quad (3)$$

So we obtain:

$$E(Y) \leq \alpha \sum_{i=1}^{n} Var(Z_i) + Pr[B]\beta \sum_{i=1}^{n} Var(Z_i) + Pr[C]E(Y|C) \quad (4)$$

Later we will prove that for fixed α and β sufficiently large we have:

$$Pr[C]E(Y|C) < \alpha \sum_{i=1}^{n} Var(Z_i) \quad (5)$$

Knowing that, we can obtain: $Pr[B] \geq \frac{1-2\alpha}{\beta} > 0$ for $\alpha < \frac{1}{2}$ Now, taking: $T = \sqrt{\frac{\alpha}{4}}$ we have:

$$Pr[|X - EX| > 2T\sqrt{\sum_{i=1}^{n} Var(Z_i)}] = Pr[Y > 4T^2 \sum_{i=1}^{n} Var(Z_i)] \quad (6)$$

Therefore

$$Pr[|X - EX| > 2T\sqrt{\sum_{i=1}^{n} Var(Z_i)}] \geq Pr[B] \geq \frac{1-2\alpha}{\beta} \quad (7)$$

So we obtain the inequality we were looking for to complete the proof in this case. The crucial thing now is to appropriately bound expression: $Pr[C]E(Y|C)$ for sufficiently large β. We need to show that for β large enough we have:

$$E(YI\{Y > \beta \sum_{i=1}^{n} Var(Z_i)\}) < \alpha \sum_{i=1}^{n} Var(Z_i) \quad (8)$$

which is equivalent to:

$$E(\frac{Y}{\sum_{i=1}^{n} Var(Z_i)} I\{\frac{Y}{\sum_{i=1}^{n} Var(Z_i)} > \beta\}) < \alpha \quad (9)$$

We introduce the following denotation:

$$D_n = \frac{\sum_{i=1}^{n} Z_i}{\sqrt{\sum_{i=1}^{n} Var(Z_i)}}$$

We would like to prove that: $E(D_n^2 I\{D_n^2 > \beta\}) < \alpha$ for β large enough.

Remark 3. *For ϕ being Gaussian or Poissonian each D_n is also Gaussian or Poissonian. Besides, each has variance equal to 1 and mean 0. So trivially, for ϕ being Gaussian or Poissonian the inequality above is satisfied for β large enough and we are done.*

Assume now that ϕ is bounded. We will bound the probability: $Pr(|D_n| > c)$ for some fixed $c > 0$. We will use the following version of Azuma's inequality (see: [She96]):

Lemma 2. *Let M_i, $i \geq 1$ be a martingale with mean $\mu = E[M_i]$. Let $M_0 = \mu$ and suppose that for nonnegative constants α_i, β_i, $i \geq 1$,*

$$-\alpha_i \leq M_i - M_{i-1} \leq \beta_i$$

. Then for any $n \geq 0, a \geq 0$:

$$Pr(M_n - \mu \geq a) \leq exp(-\frac{2a^2}{\sum_{i=1}^{n}(\alpha_i + \beta_i)^2})$$

and

$$Pr(M_n - \mu \leq -a) \leq exp(-\frac{2a^2}{\sum_{i=1}^{n}(\alpha_i + \beta_i)^2})$$

We apply lemma 2 to the sequence defined as follows:

- $M_0 = 0$
- $M_i = \sum_{i=1}^{n} Z_i$, $i = 1, 2, ..., n$

It is easy to check that $\{M_i\}$ is a martingale of mean 0. We have:

$$Pr(|D_n| > c) = Pr(|M_n| > c\sqrt{\sum_{i=1}^{n} Var(Z_i)}) \quad (10)$$

Therefore from the definition of Z_i:

$$Pr(|D_n| > c) = Pr(|M_n| > c\sqrt{Var(\phi)}\sqrt{\sum_{i=1}^{n}(x_i - d_i)^2}) \quad (11)$$

So we can apply lemma 2 to obtain:

$$Pr(|D_n| > c) \leq 2exp(-\frac{2c^2 Var(\phi)\sum_{i=1}^{n}(x_i - d_i)^2}{\sum_{i=1}^{n}(\alpha_i + \beta_i)^2}), \quad (12)$$

where $\alpha_i = \beta_i = m_\phi |x_i - d_i|$ and m_ϕ is an upper bound on $|\phi|$.

So we have:

$$Pr(|D_n| > c) \leq 2exp(-\frac{c^2 Var(\phi)}{2m_\phi^2}) \quad (13)$$

We have:

$$E(D_n^2 I\{D_n^2 > \beta\}) = \beta Pr[D_n^2 > \beta] + \int_{\beta}^{\infty} Pr[D_n^2 > y]\, dy. \quad (14)$$

So using 13, we obtain:

$$E(D_n^2 I\{D_n^2 > \beta\}) \leq 2\beta e^{-\frac{\beta Var(\phi)}{2m_\phi^2}} + 2\int_{\beta}^{\infty} e^{-\frac{Var(\phi)}{2m_\phi^2} y} dy \quad (15)$$

And this last upper bound on $E(D_n^2 I\{D_n^2 > \beta\})$ that we obtained obviously converges to 0 when $\beta \rightarrow \infty$. So we proved that for a bounded ϕ we have: $E(D_n^2 I\{D_n^2 > \beta\}) < \alpha$ for β large enough.

From what we said so far we know that if ϕ is primal then indeed: $E(D_n^2 I\{D_n^2 > \beta\}) < \alpha$ for β large enough. However for a general setting we need a little bit different approach.

From the equation 14 we see that it suffices to prove that: $\beta Pr[D_n > \sqrt{\beta}]$,$\beta Pr[D_n < -\sqrt{\beta}]$, $\int_{\beta}^{\infty} Pr[D_n > \sqrt{y}]\,dy$, $\int_{\beta}^{\infty} Pr[D_n < -\sqrt{y}]\,dy$ are all arbitrarily close to 0 when β is large enough.

To do that, we use Central Limit Theorem, and more precisely: some version of Berry-Esseen inequality, that will enable us to make some approximations using Gaussian distribution.

We use the following lemma (see: [NT07]):

Lemma 3. *Let $\{S_1, S_2, ...S_n\}$ be a sequence of independent random variables with 0 mean, not necessarily identically distributed, with finite third moment. Assume that $\sum_{i=1}^{n} E(S_i^2) = 1$. Define: $W_n = \sum_{i=1}^{n} S_i$. Then:*

$$|Pr[W_n \leq x] - \phi(x)| \leq \frac{C_1}{1+|x|^3} \sum_{i=1}^{n} E(|S_i|^3),$$

for some constant C_1 (we can take $C_1 = 30.84$), where $\phi(x) = Pr(g \leq x)$ and g is a normal distribution with variance 1 and mean 0.

We can use this lemma taking: $S_i = \frac{Z_i}{\sqrt{\sum_{i=1}^{n} Var(Z_i)}}$. Then we have: $W_n = D_n$. It is easy to check that all the conditions that are required to use lemma 3 are satisfied under such a choice. In particular, S_i chosen in such a way has finite third moment because ϕ has finite third moment.

Denote: $\bar{\phi}(x) = 1 - \phi(x)$. From lemma 3 we know that:

$$|Pr[D_n > x] - \bar{\phi}(x)| \leq \frac{C_1}{1+|x|^3} \sum_{i=1}^{n} E(\frac{|Z_i|^3}{\sqrt{\sum_{i=1}^{n} Var(Z_i)}^3}) \tag{16}$$

We also know that:

$$E(\frac{|Z_i|^3}{\sqrt{\sum_{i=1}^{n} Var(Z_i)}^3}) \leq \frac{\rho}{\sqrt{\frac{h^6(n)}{g^6(n)}}} \tag{17}$$

for some constant ρ.

That is true because $\sup_{i \in \{1,2,...n\}} E(|Z_i|^3)$ is finite and

$$\sum_{i=1}^{n} Var(Z_i) \geq \sigma \frac{h^2(n)}{g^2(n)}$$

for some constant σ.

As an immediate corollary we have:

$$\sum_{i=1}^{n} E|S_i^3| = O(\frac{n}{\sqrt{\frac{h^6(n)}{g^6(n)}}}) = o(1) \tag{18}$$

because of the assumption that we put on g in the statement of the disqualifying lemma. So we have:

$$\beta Pr[D_n > \sqrt{\beta}] \leq \beta(\bar{\phi}(\sqrt{\beta}) + \frac{C}{1+\beta^{\frac{3}{2}}}) \tag{19}$$

for some positive constant C.

But it is easy to check that expression on the right, bounding $\beta Pr[D_n > \sqrt{\beta}]$, tends to 0 as $\beta \to \infty$.

So we have:

$$\lim_{\beta \to \infty} \sup_{n} \beta Pr[D_n > \sqrt{\beta}] = 0 \tag{20}$$

Similarly:

$$\lim_{\beta \to \infty} \sup_{n} \beta Pr[D_n < -\sqrt{\beta}] = 0 \tag{21}$$

This inequality can be obtained by taking $S_i = -\frac{Z_i}{\sqrt{\sum_{i=1}^{n} Var(Z_i)}}$ in Berry-Esseen inequality and using the same trick as before.

We also have:

$$\int_{\beta}^{\infty} Pr[D_n > \sqrt{y}]\,dy \leq \int_{\beta}^{\infty} (\bar{\phi}(\sqrt{y}) + \frac{C}{1+y^{\frac{3}{2}}})\,dy \tag{22}$$

And again it is easy to check that expression on the right above converges to 0 as $\beta \to \infty$.

So we also have:

$$\lim_{\beta \to \infty} \sup_{n} \int_{\beta}^{\infty} Pr[D_n > \sqrt{y}]\,dy = 0 \tag{23}$$

And by the same analysis we also get:

$$\lim_{\beta \to \infty} \sup_{n} \int_{\beta}^{\infty} Pr[D_n < -\sqrt{y}]\,dy = 0 \tag{24}$$

Therefore we showed that:

$$\lim_{\beta \to \infty} \sup_{n} E(D_n^2 I\{D_n^2 > \beta\}) = 0 \tag{25}$$

This is all that we need to prove disqualifying lemma for the first case. Remember that we haven't chosen parameter α yet. So far we only needed: $0 < \alpha < \frac{1}{2}$. When α is fixed, parameter T is determined and we have in fact: $T = \sqrt{\frac{\alpha}{4}}$. We will now consider second case when $|E(X)| > T\sqrt{\sum_{i=1}^{n} Var(Z_i)}$. We will finally determine α and, as a consequence, T.

A.2.2 Case 2: $|E(X)| > T\sqrt{\sum_{i=1}^{n} Var(Z_i)}$

We assume that $E(X) > 0$. For $E(X) < 0$ the proof is analogous. Observe that it is enough to prove that: $Pr[(X - EX) < -\gamma\sqrt{\sum_{i=1}^{n} Var(Z_i)}] < \delta$ for some $\delta < 1$ and $0 < \gamma < T$. This follows from the fact that $\epsilon = o(\sqrt{\sum_{i=1}^{n} Var(Z_i)})$. So in fact we'd like to bound: $Pr[D_n < -\gamma]$. For ϕ being Gaussian or Poissonian, from what we have said so far, each D_n is also Gaussian or Poissonian with mean 0 and variance equal to 1. So we are done in this case. For ϕ being bounded, using the same analysis as in Case 1, we can easily obtain for any $\gamma > 0$: $Pr[D_n < -\gamma] \leq exp(-\frac{\gamma^2 Var(\phi)}{2m_{\phi}^2}) < 1$. So we are also done in this case. For the general setting, with our additional condition on $g(n)$, finding an upper bound for $Pr[D_n < -\gamma]$ is easy too. The latter probability is equal to $Pr[-D_n > \gamma]$ so we can use Berry-Esseen inequality, taking: $S_i = -\frac{Z_i}{\sum_{i=1}^{n} Var(Z_i)}$. Thus we obtain:

$$Pr[D_n < -\gamma] \leq \bar{\phi}(\gamma) + \frac{C_2}{1+\gamma^3} \sum_{i=1}^{n} E(|S_i|^3) \tag{26}$$

for some constant C_2. For $g(n) = o(n^{-\frac{1}{3}} h(n))$ and n sufficiently large, the expression $\frac{C_2}{1+\gamma^3} \sum_{i=1}^{n} E(|S_i|^3)$ is arbitrarily small. So taking for example $\alpha = \frac{1}{3}$ and $\gamma = \sqrt{\frac{1}{13}}$ we can find n_0 such that

$$\forall_{n > n_0} \frac{C_2}{1+\gamma^3} \sum_{i=1}^{n} E(|S_i|^3) < \phi(\gamma)$$

□