New Efficient Attacks on Statistical Disclosure Control Mechanisms

Cynthia Dwork and Sergey Yekhanin

Microsoft Research
{dwork,yekhanin}@microsoft.com

Abstract. The goal of a statistical database is to provide statistics about a population while simultaneously protecting the privacy of the individual records in the database. The tension between privacy and usability of statistical databases has attracted much attention in statistics, theoretical computer science, security, and database communities in recent years. A line of research initiated by Dinur and Nissim investigates for a particular type of queries, lower bounds on the distortion needed in order to prevent gross violations of privacy. The first result in the current paper simplifies and sharpens the Dinur and Nissim result.

The Dinur-Nissim style results are strong because they demonstrate insecurity of all low-distortion privacy mechanisms. The attacks have an all-or-nothing flavor: letting n denote the size of the database, $\Omega(n)$ queries are made before anything is learned, at which point $\Theta(n)$ secret bits are revealed. Restricting attention to a wide and realistic subset of possible low-distortion mechanisms, our second result is a more acute attack, requiring only a fixed number of queries for each bit revealed.

1 Introduction

The goal of a statistical database is to provide statistics about a population while simultaneously protecting the privacy of the individual records in the database. A natural example that highlights the tension between usability and preserving privacy is a hospital database containing medical records of the patients. On one hand, the hospital would like allow medical research that is based on the information in the database. On the other hand, the hospital is legally obliged to protect the privacy of its patients, i.e., leak no information regarding the medical condition of any specific patient that can be "traced back" to the individual.

The tension between privacy and usability of statistical databases has attracted considerable attention in the statistics, theoretical computer science, cryptography, security, and database communities since late 1970s. There is a a vast body of work on this subject (for references, see [1, 6, 10, 24, 25, 26]). However, the formal treatment of privacy has generally been unsatisfactory, either due to lack of specificity or because the notion of privacy compromise was not sufficiently general to capture many forms of leakage that ordinary people would still find unacceptable, or the schemes ensure security only against certain specific class of attacks.

D. Wagner (Ed.): CRYPTO 2008, LNCS 5157, pp. 469–480, 2008.

In a seminal paper [12] Dinur and Nissim initiated a rigorous study of the tradeoff between privacy and usability. They focused on a class of techniques that Adam and Wortmann, in their encyclopedic 1989 survey [1] of statistical disclosure control methods, call *output perturbation.* Roughly speaking, a query is a function that maps the database to a (real) number, and an output perturbation statistical disclosure control mechanism (curator) simply adds noise to the answers. Thus, the true answer of (say) 1910, may be reported as 1914 or 1907. The degree of distortion, that is, the magnitude of the noise, is an important measure of the utility of the mechanism. Dinur and Nissim formulated and investigated the question of how large the noise magnitude needs to be in order to ensure privacy in the following setting: each of the n rows in the database is a single bit, a query is specified by naming a subset $S \subseteq [n]$ of the rows, and the true answer to the query is the number of 1's in the specified set of rows: $\sum_{i \in S} d_i$, where d_i is the bit in the ith row, $1 \leq i \leq n$. They demonstrated a powerful attack on database curators, and concluded that *every* database curator that gives too accurate answers to too many queries inherently leaks private information [12].

The negative results of [12] have been highly influential. On one hand, they were a catalyst for a fruitful direction of obtaining provably secure statistical disclosure control mechanisms [3, 4, 9, 12, 16, 18, 20, 21, 23]. Provably secure mechanisms are now available [3, 9, 16, 18] for many standard data-mining tasks such as singular value decomposition, k-means clustering, principal component analysis, the perceptron algorithm, and contingency table release, several tasks in learning theory [4, 9, 20], and distributed implementations of these [15]. All these mechanisms have the property, shown to be inherent by [12], the magnitude of the noise increases with the number of questions asked. On the other hand, the results of Dinur and Nissim are important, unexpected, and rather disappointing for many research statisticians who often assumed, or at least hoped, that privacy can be achieved via a hypothetical clever "one-shot" procedure, that would turn the database into a sanitized object, permitting significantly accurate answers to be derived for queries that are not specified on the outset, without a risk of privacy violation.

In this paper we continue the line of research initiated by Dinur and Nissim. Our first result in the current paper simplifies and sharpens the Dinur and Nissim result. Our second result shows a limitation for a type of privacy mechanisms that includes tabular data release and synthetic data sets. We show a class of queries for which even adding *arbitrary* noise to a $(1/2-\epsilon)$ fraction of the answers fails to protect privacy against an adversary running in time independent of the database size. Thus, no mechanism of the specified type can safely provide very accurate answers to even a $(1/2+\epsilon)$ fraction of these queries.

Before stating our contributions formally we discuss the results of Dinur and Nissim [12] and Dwork et al. [17] in more detail.

1.1 Results and Earlier Work

In an interactive privacy mechanism the database curator is a trusted entity that sits between the possibly adversarial user of the database and the actual data. Given a query, the curator computes the correct answer and adds some noise to the response. When the database is a vector of bits a mechanism is *blatantly non-private* if, after interacting with a database curator, an adversary can produce a candidate database c that agrees with the real database on all but $o(n)$ entries, i.e., $d_i = c_i$ for all but $o(n)$ values of $1 \leq i \leq n$. This model, while at first blush simplistic, is in fact sufficiently rich to capture many natural questions. A detailed discussion of the model can be found in [12, 14, 17].

Dinur and Nissim [12] showed that a mechanism in which curator that adds $o(\sqrt{n})$ noise to every response is blatantly non-private against a polynomial-time bounded adversary asking $O(n \log^2 n)$ questions[1].

At a high level, the attack of [12] proceeds in two steps. In the first step the adversary poses $O(n \log^2 n)$ random subset-sum queries, chosen by including each database record uniformly and independently with probability 1/2. In the second step the adversary solves a linear system of equations with n variables and $O(n \log^2 n)$ constraints in order to find a candidate database that fits all available data. The second step of the attack carries most of the computational burden. The most efficient linear programming algorithm to date is due to Pravin M. Vaidya [27]. It requires $O(((m+n)n^2 + (m+n)^{1.5}n)L)$ arithmetic operations where m is the number of constraints, n is the number of variables, and L is bounded by the number of bits in the input. In the setting of [12] this yields an $O(n^5 \log^4 n)$ worst case running time.

Our first result sharpens the Dinur-Nissim attack. The new attack requires only n deterministically chosen queries, requires significantly less computation. Also of value, our analysis is much simpler, relying only on basic properties of the Fourier transform over the group $\mathbb{Z}_2^k$.

The key message of the Dinur-Nissim work is that any database curator that gives reasonably accurate answers to too many queries leaks private information. This however leaves open a possibility that some curator giving wildly inaccurate answers to a (small) fraction of the queries, and reasonably accurate answers to the rest may preserve privacy. Existence of such curators was studied by Dwork et al. [17], who have showed that if the query is now a vector of n standard normals, and the true answer is the inner product of the database with the vector, then any database mechanism adding noise bounded by $o(\sqrt{n})$ to at least 0.761 fraction of its responses is blatantly non-private[2]. Inspired by the LP decoding methods from the literature on compressed sensing of signals, e.g. [5, 7, 8], this attack also requires solving a random linear program with n variables and $O(n)$ constraints, and so has a worst case running time of $O(n^5)$.

Although the actual constant (≈ 0.761) is shown to be sharp threshold for LP decoding [17], other attacks may be possible. Indeed, it is plausible that every statistical disclosure control mechanism that adds low noise to $(1/2 + \epsilon)$

[1] Dinur and Nissim also showed that if the adversary can ask 2^n queries then the mechanism is blatantly non-private as long as the noise is magnitude is $o(n)$; however, here we restrict our attention to efficient adversaries.

[2] We think of this as a natural generalization of the Dinur-Nissim attack, in which the query vector is restricted to having binary entries.

fraction of its responses (and allows for sufficiently many queries) leaks private information, for any $\epsilon > 0$. Dwork et al. [17] have made a step towards proving this claim. Namely, they came up with an inefficient (i.e., $\exp(n)$-time) adversary that asks $O(n)$ questions from the curator and achieves blatant nonprivacy, in case the curator gives reasonably accurate responses to $(1/2 + \epsilon)$ fraction of queries.[3]

In our second result we address the question of whether a mechanism that adds unbounded noise to a $(1/2 - \epsilon)$ fraction of its responses can ensure privacy, and prove the contrary for a certain range of parameters. We obtain an attack running in $\text{poly}(e/\epsilon)$ time that can tolerate the optimal $(1/2 - \epsilon)$ fraction of unbounded noise provided the noise on the rest of the queries is at most e. As in the case of the previous attacks, the query is a vector of length n, and the true answer is the inner product of the (binary) database and the query vector. Note that the running time is independent of n; we are not counting the time needed to formulate the query (not hard, but depends on n) and to compute the response.

Note that one needs to be careful when specifying a fraction of queries to which a certain curator adds unbounded (or low) noise since curators can be of very different nature. In particular some curators may allow only for a certain (small) number of queries, and some may give different answers to the same query asked two times in a row.

Our attack applies to database curators that for certain values of p, given a (randomized) p-sized collection of queries coming from a 2-independent family add low noise to $(1/2 + \epsilon)p$ of their responses with a probability bounded away from $1/2$.

The class of curators that fall into the above category is quite broad. Specifically (as we later prove) it includes curators that may study the database and release an "object" that the adversary/analyst can study as desired. This captures, for example, completely non-interactive solutions such as: summary tables, statistics, synthetic data sets, and all other known forms of statistical data release in use today, but it also includes (hypothetical) programs for handling certain classes of queries with obfuscated data hard-wired in. Our model also (obviously) captures interactive curators (i.e., curators that keep a query log and adjust their responses to incoming queries based on such a log) that allow for p queries, and add unbounded noise to at most $(1/2 - \epsilon)p$ responses.

Our attack has important differences from the earlier attacks in the literature:

One Bit at a Time. Conceptually, our adversary attacks one bit at a time. That is, the adversary chooses a bit to attack and runs a simple program in which it forms queries and interacts with a database curator in order to obtain a

[3] Note that there are database curators that reveal no information about specific database records and give correct answers to exactly $1/2$ of the queries. For instance, consider a database curator that answers one half of the queries according to a database x and the other half of the queries according to a complement database $\bar{x}$. Clearly, an interaction with such a curator will not help an adversary distinguish an all-zeros database from an all-ones database.

(potentially wildly) noisy version of the true answer. The adversary can increase its success probability by running the attack multiple times. The adversary can attack the entire database by running the attack n times.

Small Noise is Very Small. The magnitude of the noise on the $(1/2+\epsilon)$ fraction of "good" responses will be bounded by something *smaller than* the maximum allowable coefficient in the query vector. This is the weakest aspect of our result. However, prior to this work no efficient attack was known even when a $(1/2+\epsilon)$ fraction of the responses have *zero* noise.

Viewed differently, the result says that if the "good" responses must, for reasons of utility, have noise bounded by some number p, then the system cannot safely permit $O(p)$ subset sum queries with coefficients even as large as $2p+1$.

Our attack is based on a new interplay between the basic properties of polynomials over reals and ideas coming from the theory of error-correcting codes [22].

2 Preliminaries

We start with the basic definitions. A database for us is simply an n-bit string $d = (d_1, \ldots, d_n) \in \{0,1\}^n$.

Definition 1. *Let d be an n-bit database. A query is a vector $q \in \mathbb{R}^n$. The* true answer *to a query q is the inner product $q \cdot d$, i.e., the weighted sum of database bits $a_q = \sum_{i \in q} q_i d_i$. A disclosure control mechanism $\mathcal{C}$ takes as input a query q and database d and provides a possibly noisy response in $\mathbb{R}$, for which $\mathcal{C}$ may employ randomness. We say that a response $\mathcal{C}(x,q)$ carries noise of at most σ if $|\mathcal{C}(x,q) - a_q| \leq \sigma$.*

The following formalization of *non-privacy*, due to Dinur and Nissim [12], has come to be called *blatant non-privacy.*

Definition 2. *Let $\mathcal{C}$ be a privacy mechanism. We say that $\mathcal{C}$ is blatantly non-private against a probabilistic algorithm $\mathcal{A}$ (an adversary) if after an interaction with $\mathcal{C}$, $\mathcal{A}$ recovers most of the database d with very high probability. Formally, for all $d \in \{0,1\}^n$,*

$$Pr[\mathcal{A}^{\mathcal{C}} \mathit{outputs}\ y \in \{0,1\}^n \ \mathit{such\ that}\ d_H(d,y) \in o(n)] \geq 1 - neg(n),$$

where the probability is taken over the randomness of $\mathcal{A}$, $neg(n)$ denotes a function that is asymptotically smaller than any inverse polynomial in n, and $d_H(x,y)$ stands for the Hamming distance.

Definition 3. *We also say that $\mathcal{C}$ is $(1-\delta)$-non-private against an adversary $\mathcal{A}$ if for an arbitrary $i \in [n]$, $\mathcal{A}$ can recover the value of the bit d_i after an interaction with $\mathcal{C}$ with probability $1-\delta$. Formally, $\forall\ d \in \{0,1\}^n$, $\forall\ 1 \leq i \leq n$,*

$$Pr[\mathcal{A}^{\mathcal{C}}(i) \mathit{generates}\ z \in \{0,1\}\ \mathit{such\ that}\ z = d_i] \geq 1 - \delta,$$

where the probability is taken over the random coin tosses of $\mathcal{C}$ and $\mathcal{A}$.

Clearly, the the definition above is useful only if $\delta < 1/2$. Note that the definition is very strong. It says that the curator $\mathcal{C}$ fails to protect *every* database record.

We measure the complexity of an attack on a statistical disclosure control mechanism with respect to: 1. the number of queries asked by an adversary; 2. the running time of an adversary. Our attacks (and all other attacks in the literature) are non-adaptive. They proceed in two steps. First an adversary asks all its questions from a curator; next the adversary processes the curator's responses in order to reveal some private information. We define the time complexity of an attack to be the time complexity of the second step.

3 Fourier Attack: $o(\sqrt{n})$ Noise, n Queries, $O(n \log n)$ Running Time

The goal of this section is to establish the following theorem.

Theorem 1. *There exists an adversary $\mathcal{A}$ that runs in $O(n \log n)$ time and achieves a blatant privacy violation against any database curator $\mathcal{C}$ that allows for n queries with integer $\{0,1\}$ coefficients and adds $o(\sqrt{n})$ noise to every response.*

Our proof of theorem 1 relies on some standard properties of the Fourier transform over the finite group $\mathbb{Z}_2^k$. In the following subsection we briefly review the properties that are important for us.

3.1 Fourier Preliminaries

Characters of $\mathbb{Z}_2^k$ are homomorphisms from $\mathbb{Z}_2^k$ into the multiplicative group $\{\pm 1\}$. There exist 2^k characters. We denote characters by χ_a, where $a = (a_1, \ldots, a_k)$ ranges in $\mathbb{Z}_2^k$, and set $\chi_a(x) = (-1)^{\sum_{i=1}^n a_i x_i}$ for every $x = (x_1, \ldots, x_k) \in \mathbb{Z}_2^k$. Let $f(x)$ be a function from $\mathbb{Z}_2^k$ into reals. For an arbitrary $a \in \mathbb{Z}_2^k$ the Fourier coefficient $\hat{f}(\chi_a)$ is defined by $\hat{f}(\chi_a) = \sum \chi_a(x) f(x)$, where the sum is over all $x \in \mathbb{Z}_2^k$. For every $a \in \mathbb{Z}_2^k$ consider a set

$$S_a = \left\{ x \in \mathbb{Z}_2^k \mid \sum_{i=1}^k a_i x_i = 0 \mod (2) \right\}. \tag{1}$$

It is easy to see that the size of S_a is equal to 2^k if $a = 0^k$, and is equal to 2^{k-1} otherwise. For every $a \in \mathbb{Z}_2^k$ consider the sum

$$\sigma_a(f) = \sum_{x \in S_a} f(x). \tag{2}$$

The Fourier coefficients of a function f can be easily expressed in terms of sums $\sigma_a(f)$:

$$\hat{f}(\chi_a) = \begin{cases} \sigma_0(f), & \text{if } a = 0; \\ 2\sigma_a(f) - \sigma_0(f), & \text{otherwise.} \end{cases} \tag{3}$$

Let $\hat{f} = (\hat{f}(\chi_a))_{a \in \mathbb{Z}_2^k}$ be a vector of Fourier coefficients of f. Consider a matrix $H \in \{\pm 1\}^{2^k \times 2^k}$. Rows and columns of H are labelled by elements of $\mathbb{Z}_2^k$ (taken in the same order). $H_{a,b} = \chi_a(b)$. H is a (Sylvester type) Hadamard matrix. It is not hard to verify that $HH = 2^k I$, where I is the identity matrix. Note that $\hat{f} = Hf$. Therefore

$$f = \frac{1}{2^k} H \hat{f}, \tag{4}$$

i.e., an inverse of a Fourier transform is simply another Fourier transform up to a scalar multiplication. The following classical (Parseval's) identity relates the absolute values of f to the absolute values of the Fourier coefficients of f :

$$\sum_{x \in \mathbb{Z}_2^k} |f(x)|^2 = \frac{1}{2^k} \sum_{a \in \mathbb{Z}_2^k} |\hat{f}(\chi_a)|^2. \tag{5}$$

3.2 The Attack

Proof of theorem 1. Let $d = (d_1, \ldots, d_n)$ be the database. Without a loss of generality assume that n is a power of two, $n = 2^k$. Consider an arbitrary bijection $g : \mathbb{Z}_2^k \to [n]$ between the group $\mathbb{Z}_2^k$ and the set $[n]$. Now database d defines a map f from $\mathbb{Z}_2^k$ to $\{0, 1\}$, where we set $f(x) = d_{g(x)}$, for every $x \in \mathbb{Z}_2^k$. Our attack proceeds in three steps. Firstly, the adversary $\mathcal{A}$ asks n queries from the curator $\mathcal{C}$ to obtain the noisy version of sums $\sigma_a(f)$. Secondly, $\mathcal{A}$ performs a simple computation to derive noisy Fourier coefficients of f from the curator's responses. Finally, $\mathcal{A}$ performs an inverse Fourier transform to (approximately) recover the function f from its noisy Fourier spectrum. Below is a more formal description.

- For every $a \in \mathbb{Z}_2^k$, $\mathcal{A}$ asks for the sum of database bits in the set S_a, where S_a is defined by formula (1). $\mathcal{A}$ obtains the noisy values $\tilde{\sigma}_a(f)$ of sums $\sigma_a(f)$. Note that for every $a \in \mathbb{Z}_2^k$ we have $\tilde{\sigma}_a(f) = \sigma_a(f) + o(\sqrt{n})$.
- $\mathcal{A}$ uses formula (3) to obtain a vector $\tilde{f}$ of noisy Fourier coefficients of f. Note that $\tilde{f} = \hat{f} + e$, where the absolute value of every coordinate of e is bounded by $o(\sqrt{n})$.
- $\mathcal{A}$ applies formula (4) to obtain a noisy version of f from $\tilde{f}$. Specifically, $\mathcal{A}$ computes $h = \frac{1}{n} H \tilde{f}$, and for every coordinate $i \in [n]$, sets $y_i = 0$ if $h_i < 1/2$ and $y_i = 1$ otherwise.

Note that there are $O(n \log n)$ time algorithms to compute Fourier transform [11]. Therefore the overall running time of the attack is $O(n \log n)$. We now argue that the attacker always recovers a database y such that $d_H(d, y) = o(n)$. The linearity of the Fourier transform implies that it would suffice for us to show that the vector $\frac{1}{n} He$, can not have $\Omega(n)$ coordinates with absolute values above 1/2. This follows immediately from the Parseval's identity (5) that asserts that the L_2 norm of e is n times larger than the L_2 norm of $\frac{1}{n} He$. □

3.3 Summary of First Result

We presented a novel attack on statistical disclosure control mechanism that applies in the model considered earlier by Dinur and Nissim [12]. We believe that the most important feature of our attack is its conceptual simplicity; in addition, it is sharper than that of Dinur and Nissim [12] in the following respects:

- Our adversary makes fewer queries (n versus $O(n \log^2 n)$).
- Both algorithms first pose queries and then analyze the results. Our analysis is computationally more efficient ($O(n \log n)$ vs $\Omega(n^5 \log^4 n)$ worst case running time).
- Our adversary always achieves blatant non-privacy; previous attacks have a negligible probability of failure.

4 Interpolation Attack: $(1/2 - \epsilon)$ Fraction of Unbounded Noise, Poly(e/ϵ) Running Time

In this section the query vectors will have integer coefficients chosen from some range $[0, \ldots, p-1]$. Our goal is to establish the following theorem.

Theorem 2. *Let p be a prime. Suppose $0 < \epsilon \leq 1/2$, $e \geq 0$, and $\delta < 1/2$ are such that*

$$2\sqrt{(p-1)/\delta} + 8e + 3 \leq 2\epsilon(p-1) \quad \textit{and} \quad e < (p-1)/8; \tag{6}$$

then any curator $\mathcal{C}$ that given a (randomized) $(p-1)$-sized collection of queries coming from a 2-independent family adds noise less than or equal to e to at least $(1/2+\epsilon)(p-1) - \sqrt{(p-1)/\delta}$ of its responses with probability $(1-\delta)$ is $(1-\delta)$-non-private against an adversary that asks $p-1$ queries and runs in $O(p^4)$ time.

We defer the discussion of the type of curators that are vulnerable to the attack above till later in this section, and we defer the proof of theorem 2 to the following subsection. Below we state some of the immediate corollaries of the theorem. The next corollary captures the asymptotic parameters of the attack from theorem 2. To obtain it, one simply needs to set $\delta = 1/4$ and use crude estimates for p to satisfy (6).

Corollary 1. *Let $0 < \epsilon \leq 1/2$ and $e \geq 0$ be arbitrary. Let p be a prime such that $p \geq 20/\epsilon^2$ and $p \geq 15e/\epsilon$. Suppose a database curator $\mathcal{C}$ allows queries with integer weights from $[0, \ldots, p-1]$. Also, assume that given a (randomized) $(p-1)$-sized collection of queries coming from a 2-independent family $\mathcal{C}$ adds noise less than or equal to e to at least $(1/2+\epsilon)(p-1) - \sqrt{(p-1)/\delta}$ of its responses with probability $3/4$. Then $\mathcal{C}$ is $3/4$-non-private against an adversary that issues $O(p)$ queries and runs in $O(p^4)$ time.*

The corollary above may be somewhat unsatisfying since the adversary has a substantial (1/4) probability of failing to correctly reveal private information. Note however, that (assuming the curator allows for more queries) the adversary can run its attack multiple times, to obtain independent estimates $y_i^{(1)}, \ldots, y_i^{(t)}$ for a ceratin specific bit d_i of the database d. Next the adversary can report the majority of $\{y_i^{(j)}\}_{j \in [t]}$ as a new estimate for d_i. A standard argument based on the Chernoff bound [2] shows that the new estimate has a vastly lower probability of an error.

Corollary 2. *Let $0 < \epsilon \leq 1/2$ and $e \geq 0$ be arbitrary. Let p be a prime such that $p \geq 20/\epsilon^2$ and $p \geq 15e/\epsilon$. Suppose a database curator $\mathcal{C}$ allows queries with integer weights from $[0, \ldots, p-1]$. Also, assume that given a (randomized) $(p-1)$-sized collection of queries coming from a 2-independent family $\mathcal{C}$ adds noise less than or equal to e to at least $(1/2+\epsilon)(p-1) - \sqrt{(p-1)/\delta}$ of its responses with probability 3/4. Then for every integer $t \geq 1$, $\mathcal{C}$ is $(1-2^{-t/12})$-non-private against an adversary that issues $O(tp)$ queries and runs in $O(tp^4)$ time.*

We now argue that the condition of theorem 2 (and corollaries 1 and 2) holds for non-interactive database curators whose responses to more than $(1/2+\epsilon)$ fraction of all possible queries carry low noise. Our argument relies on the following lemma that gives a well-known property of pairwise independent samples. The lemma follows from the fact that for pairwise independent random variables, the variance is the sum of the variances, and the Chebychev's inequality [19, lemma 2].

Lemma 1. *If S is a pairwise independent sample of elements from some domain D and I maps elements of D to the range $\{0,1\}$; then for any $\delta > 0$,*

$$\Pr\left[\left|\frac{\sum_{\mathrm{x}\in\mathrm{S}} \mathrm{I(x)}}{|\mathrm{S}|} - \mathrm{E[I(x)]}\right| \geq 1/\sqrt{\delta|\mathrm{S}|}\right] \leq \delta.$$

Let $\mathcal{C}$ be a database curator such that $\mathcal{C}$'s responses to more than $(1/2+\epsilon)$ fraction of all possible queries carry low noise. Let D be the domain of all possible queries and $I(w) : \{0,1\}^n \rightarrow \{0,1\}$ to be the incidence function of the set of queries that carry unbounded noise according to $\mathcal{C}$. Clearly, $E[I(x)] \leq 1/2 - \epsilon$. Therefore lemma 1 implies that with probability at least $1-\delta$ the total number of points that carry unbounded noise in a random sample S of size $p-1$ is at most $(1/2-\epsilon)(p-1) + \sqrt{(p-1)/\delta}$ and theorem 2 applies.

We note that theorem 2 is weak in that the small noise is very small – considerably less than the maximum allowable coefficient in a query. In fact, this noise model even rules out a privacy mechanism that protects a single bit, say, d_i, by "flipping" it – replacing d_i with its complement $1-d_i$, and then answering all queries accurately thereafter. On the other hand, to our knowledge, this is the first efficient adversary that successfully attacks any mechanism that can add arbitrary noise in a $(1/2-\epsilon)$ fraction of the responses.

4.1 The Attack

The main idea behind the proof of Theorem 2 is to achieve error-correction via polynomial interpolation. This idea has been previously extensively used (in a related, yet distinct setting) of local decoding of Reed-Muller codes [19, 22]. Below is a high-level overview of our attack.

The attacker $\mathcal{A}$ thinks of its queries as points $q = (q_1, \dots, q_n) \in \mathbb{F}_p^n$ in an n-dimensional linear space over a prime field $\mathbb{F}_p$. $\mathcal{A}$ reduces all responses to its queries modulo p, and treats the database $d = (d_1, \dots, d_n) \in \{0,1\}^n$ as an unknown n-variate linear form $f(q_1, \dots, q_n) = \sum_{i=1}^{n} d_i q_i$ over $\mathbb{F}_p$. It is easy to see that in order to recover (say) the first bit of d, it would suffice for $\mathcal{A}$ to determine the value of $f(q)$ for $q = ((p-1)/2, 0, \dots, 0)$ with an error of less than $(p-1)/4$.

The attacker does not directly ask the curator for the value of $f(q)$, since the response to the query q may carry unbounded noise (and therefore be misleading), but rather issues a randomized collection of $(p-1)$ queries $q^{(1)}, \dots, q^{(p-1)}$ such that the value of $f(q)$ can (with high probability) be deduced from curator's responses to $q^{(1)}, \dots, q^{(p-1)}$ up to a small error.

Below is the formal description and analysis of our attack. The attacker's goal is to obtain an approximately correct answer to $q = ((p-1)/2, 0, \dots, 0)$ and thus recover the first bit of the database.

Proof of theorem 2

- $\mathcal{A}$ picks $u, v \in \mathbb{F}_p^n$ uniformly at random, and considers the parametric degree two curve $\chi = \{q + tu + t^2 v \mid t \in [1, \dots, p-1]\}$ in the space $\mathbb{F}_p^n$ through the point q. Note that the points of χ form a pairwise independent sample of $\mathbb{F}_p^n$. A proof of this standard fact can be found for instance in [19, claim 1]. The condition of the theorem implies that with probability at least $1 - \delta$ the total number of points that carry unbounded noise on χ is at most $(1/2 - \epsilon)(p-1) + c\sqrt{p-1}$.
- $\mathcal{A}$ issues $p-1$ queries $\left\{q^{(t)} = q + tu + t^2 v\right\}_{t \in [1, \dots, p-1]}$ corresponding to points of χ. Let $R = (r_1, \dots, r_{p-1})$ be a sequence of curator's responses to those queries reduced modulo p. In what follows we assume the attacker $\mathcal{A}$ is lucky and at most $(1/2-\epsilon)(p-1)+c\sqrt{p-1}$ responses $\{r_t\}_{t \in [p-1]}$ carry unbounded noise.

  We say that $\alpha \in \mathbb{F}_p$ is e-small if either $\alpha \in [-e, \dots, 0, \dots, e] \mod (p)$.

  We say that a polynomial $h(t) \in \mathbb{F}_p[t]$ *fits* the sequence R if $(h(t) - r_t)$ is e-small for $(1/2 + \epsilon)(p-1) - c\sqrt{p-1}$ values of t. Note that the degree two polynomial $g(t) = f(q + tu + t^2 v) \in \mathbb{F}_p[t]$, that is a restriction of the linear function f to a degree two curve χ fits R. Also note that $g(0) = f(q)$ is 0 if the first bit of the database x is zero, and $(p-1)/2$ otherwise.

  We now argue that $g(t)$ is the only polynomial that fits R up to a $2e$-small additive factor. To see this suppose that some other polynomial $g_1(t)$ also fits R; then the polynomial $g(t) - g_1(t)$ has to take $2e$-small values at

$$(p-1) - 2((1/2 - \epsilon)(p-1) + c\sqrt{p-1}) = 2\epsilon(p-1) - 2c\sqrt{p-1} \geq 8e + 3$$

points in $\mathbb{F}_p^*$, where the inequality above follows from (6). Since a (non-constant) quadratic polynomial can take the same value in at most two points and $2(4e+1) < 8e+3$ we conclude that $g(t) - g_1(t)$ is a $2e$-small constant in $\mathbb{F}_p$.

- $\mathcal{A}$ applies a brute-force search over all quadratic polynomials $g_1(t) \in \mathbb{F}_p[t]$ to find a polynomial that fits the sequence R. $\mathcal{A}$ computes the value $g_1(0)$ and outputs 0 if $g_1(0)$ is $2e$-small, and 1 otherwise. According to (6), $2e < (p-1)/4$ and therefore $\mathcal{A}$ correctly recovers the first bit of the database.

Observe that the running time of the attack is $O(p^4)$. The attack involves a brute-force search over all $O(p^3)$ quadratic polynomials in $\mathbb{F}_p[t]$ and it takes $O(p)$ time to verify if a polynomial fits the sequence R. □

4.2 Summary of Our Second Result

We presented a novel, efficient, attack on statistical disclosure control mechanisms with several unique features. The most novel feature of the attack is the use of polynomial interpolation in this context. The most interesting consequence of the attack is that it succeeds against privacy mechanisms that add unbounded noise to up to $(1/2 - \epsilon)$ fraction of their responses, provided the noise on other responses is sufficiently low; indeed, it even tolerates a small amount of noise in the remaining $(1/2 + \epsilon)$ responses. No efficient attacks with such property have been known earlier.

Acknowledgement

The authors thank Madhu Sudan for helpful discussions regarding the polynomial interpolation attack, and for providing the reference [19].

References

1. Adam, N., Wortmann, J.: Security-control methods for statistical databases: a comparative study. ACM Computing Surveys 21(4), 515–556 (1989)
2. Alon, N., Spencer, J.: The probabilistic method, 2nd edn. Wiley-Interscience [John Wiley and sons], New York (2000)
3. Barak, B., Chaudhuri, K., Dwork, C., Kale, S., McSherry, F., Talwar, K.: Privacy, accuracy, and consistency too: a holistic solution to contingency table release. In: Proc. of the 26th Symposium on Principles of Database Systems (PODS), pp. 273–282 (2007)
4. Blum, A., Ligett, K., Roth, A.: A learning theory approach to non-interactive database privacy. In: Proc. of the Symp. on the Theory of Computation (STOC) (2008)
5. Candes, E., Tao, T.: Near-optimal signal recovery from random projections: universal encoding strategies. IEEE Trans. Inform. Theory 52, 5406–5425 (2004)
6. Chawla, S., Dwork, C., McSherry, F., Smith, A., Wee, H.: Toward privacy in public databases. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 363–385. Springer, Heidelberg (2005)

7. Chen, S., Donoho, D., Saunders, M.: Atomic decomposition via basis pursuit. SIAM Journal on Scientific Computing 48(1), 33–61 (1999)
8. Donoho, D., Johnstone, I.: Minimax estimation via wavelet shrinkage. Annals of Statistics 26(3), 879–921 (1998)
9. Blum, A., Dwork, C., McSherry, F., Nissim, K.: Practical privacy: the SuLQ framework. In: Proc. of the 24th Symposium on Principles of Database Systems (PODS), pp. 128–138 (2005)
10. Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., Zhu, M.: Tool for privacy preserving data minining. SIGKDD Explorations 4(2), 28–34 (2002)
11. Cormen, T., Leiserson, C., Rivest, R., Stein, C.: Introduction to algorithms. MIT Press, Cambridge (2001)
12. Dinur, I., Nissim, K.: Revealing information while preserving privacy. In: Proc. of the 22nd Symposium on Principles of Database Systems (PODS), pp. 202–210 (2003)
13. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006)
14. Dwork, C.: Ask a better question, get a better answer: a new approach to private data analysis. In: Schwentick, T., Suciu, D. (eds.) ICDT 2007. LNCS, vol. 4353, pp. 18–27. Springer, Heidelberg (2006)
15. Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M.: Our data, ourselves: privacy via distributed noise generation. In: Vaudenay, S. (ed.) EUROCRYPT 2006. LNCS, vol. 4004, pp. 486–503. Springer, Heidelberg (2006)
16. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Callibrating noise to sensitivity in private data analysis. In: Halevi, S., Rabin, T. (eds.) TCC 2006. LNCS, vol. 3876, pp. 265–284. Springer, Heidelberg (2006)
17. Dwork, C., McSherry, F., Talwar, K.: The price of privacy and the limits of LP decoding. In: Proc. of the 39th Symposium on the Theory of Computation (STOC), pp. 85–94 (2007)
18. Dwork, C., Nissim, K.: Privacy preserving data-mining on vertically partitioned databases. In: Franklin, M. (ed.) CRYPTO 2004. LNCS, vol. 3152, pp. 528–544. Springer, Heidelberg (2004)
19. Gemmell, P., Sudan, M.: Highly resilient correctors for polynomials. Information Processing Letters 43(4), 169–174 (1992)
20. Kasiviswanathan, S., Lee, H., Nissim, K., Raskhodnikova, S., Smith, A.: What Can We Learn Privately? (manuscript, 2007)
21. McSherry, F., Talwar, K.: Mechanism Design via Differential Privacy. In: Proc. of the 48th Symposium on the Foundations of Computer Science (FOCS) (2007)
22. MacWilliams, F., Sloane, N.: The theory of error-correcting codes. North-Holland, Amsterdam (1977)
23. Nissim, K., Raskhodnikova, S., Smith, A.: Smooth sensitivity and sampling in private data analysis. In: Proc. of the 39th Symposium on the Theory of Computation (STOC), pp. 75–84 (2007)
24. Slavkovic, A.: Statistical disclosure limitation beyond the margins: characterization of joint distributions for contingency tables. Ph.D. thesis, Department of statistics, Carnegie Mellon University (2004)
25. Shoshani, A.: Statistical databases: Characteristics, problems and some solutiuons. In: Proc. of the 8th International Conference on Very Large Databases (VLDB), pp. 208–222 (1982)
26. Sweeney, L.: Privacy-enchanced linking. SIGKDD Explorations 7(2), 72–75 (2005)
27. Vaidya, P.: An algorithm for linear programming which requires $O(((m+n)n^2 + (m+n)^{1.5}n)L)$ arithmetic opertaions. Mathematical Programming 47, 175–201 (1990)