On the Difficulties of Disclosure Prevention in Statistical Databases or The Case for Differential Privacy

Cynthia Dwork[*] Moni Naor[†]

August 31, 2008

Abstract

In 1977 Tore Dalenius articulated a desideratum for statistical databases: *nothing about an individual should be learnable from the database that cannot be learned without access to the database.* We give a general impossibility result showing that a natural formalization of Dalenius' goal cannot be achieved if the database is useful. The key obstacle is the *side information* that may be available to an adversary. Our results hold under very general conditions regarding the database, the notion of privacy violation, and the notion of utility.

Contrary to intuition, a variant of the result threatens the privacy even of someone not in the database. This state of affairs motivated the notion of *differential privacy* [15, 16], a strong *ad omnia* privacy which, intuitively, captures the increased risk to one's privacy incurred by participating in a database.

1 Introduction

A privacy-preserving statistical database enables the data analyst to learn properties of the population as a whole, while protecting the privacy of the individuals whose data were used in creating the database. A rigorous treatment of privacy requires definitions: What constitutes a failure to preserve privacy? What is the power of the adversary whose goal it is to compromise privacy? What auxiliary information is available to the adversary (newspapers, medical studies, labor statistics) even without access to the database in question? Of course, utility also requires formal treatment, as releasing no information, or only random noise, clearly does not compromise privacy; we will return to this point later. However, in this work *privacy* is paramount: we will first define our privacy goals and then explore what utility can be achieved given that the privacy goals will be satisfied.

A 1977 paper of Dalenius [9] articulated the following desideratum: *access to a statistical database should not enable one to learn anything about an individual that could not be learned without access.* We show this type of privacy cannot be achieved in general. The obstacle is in the *auxiliary information*, that is, information available to the adversary other than from access to the statistical database. Very roughly, our main result says that in any 'reasonable' setting

[*] Microsoft Research, Silicon Valley Campus, 1065 La Avenida Mountain View, CA. 94043 USA; email: dwork@microsoft.com

[†] Incumbent of the Judith Kleeman Professorial Chair, Department of Computer Science and Applied Math, The Weizmann Institute of Science, Rehovot 76100, Israel; email: moni.naor@weizmann.ac.il. Work partly done when visiting Microsoft Research. Research supported in part by a grant from the Israel Science Foundation.

there is a piece of information that is in itself innocent, yet in conjunction with even a modified (noisy) version of the data yields a privacy breach. The intuition behind the proof of impossibility is captured by the following parable. Suppose one's exact height were considered a sensitive piece of information, and that revealing the exact height of an individual were a privacy breach. Assume that the database yields the average heights of women of different nationalities. An adversary who has access to the statistical database and the auxiliary information "Terry Gross is two inches shorter than the average Lithuanian woman" learns Terry Gross' height, while anyone learning only the auxiliary information, without access to the average heights, learns relatively little.

Related Work. There is an enormous literature on privacy in databases; we briefly mention a few fields in which the work has been carried out. See [1] for a survey of many techniques developed prior to 1989.

By far the most extensive treatment of disclosure limitation is in the statistics community; for example, in 1998 the *Journal of Official Statistics* devoted an entire issue to this question. This literature contains a wealth of privacy supportive techniques and investigations of their impact on the statistics of the data set. Rigorous definitions of privacy and modeling of the adversary are not prominent features of this portion of the literature.

Research in the theoretical computer science community in the late 1970's had very specific definitions of privacy compromise, or what the adversary must achieve to be considered successful (see, *e.g.*, [12]). The consequent privacy guarantees would today be deemed insufficiently general, as modern cryptography has shaped our understanding of the dangers of the leakage of partial information. Privacy in databases was also studied in the security community. Although the effort seems to have been abandoned for over two decades, the work of Denning [10] is closest in spirit to the line of research recently pursued in [17, 4, 16].

The work of Agrawal and Srikant [2] and the spectacular privacy compromises achieved by Sweeney [33] rekindled interest in the problem among computer scientists, particularly within the database community.

Rigorous definitions of (non)-privacy and information leakage are the pillars of modern cryptography (see, *e.g.*, [24, 25, 22]). Readers familiar with semantic security – roughly, everything computable about the plaintext given the ciphertext should be computable without access to the ciphertext [24] – will note the similarity to Dalenius' goal. Also relevant might be the definition of privacy in the context of secure function evaluation. Informally, secure function evaluation permits a set of *players* to compute a specific function $f(x_1, \ldots, x_n)$, where each x_i is the input to f provided by the ith player; the privacy guarantee is that no information is leaked about the individual values other than that which can be deduced from the output [21] (see [20] for a comprehensive treatment). In our work, privacy is paramount; in this respect it differs from the work on secure function evaluation, where privacy is ensured only modulo the function to be computed: if the function is inherently disclosive, then the privacy of secure function evaluation is meaningless. In contrast, we are concerned with *which* functions should be computed, given that privacy is mandated.

1.1 Private Data Analysis: The Setting

There are two natural models for privacy mechanisms: interactive and non-interactive. In the non-interactive setting the (trusted and trustworthy) data curator, or collector publishes a "sanitized" version of the collected data; the literature uses terms such as "anonymization" and "deidentification". Traditionally, sanitization employs techniques such as data perturbation and subsampling, as well as removing well-known identifiers such as names, birth dates, and social security numbers. It may also include releasing various types of synopses and statistics. In the interactive setting the data curator, again trusted, provides an interface through which users may pose queries about the data, and get (possibly noisy) answers. The results in this paper apply to both settings.

2 Impossibility of Absolute Disclosure Prevention

We now show that for any useful mechanism and any non-trivial notion of disclosure, there exists auxiliary information that an adversary might possess in the context of which the output of the mechanism would be disclosive.

This result should not be taken to imply that the study of disclosure limitation should be abandoned. We only demonstrate the existence of a single, arguably contrived, adversary who is configured to learn from a given non-trivial mechanism. Instead, the result should be taken to show that the prevention of disclosures is simply not a property that a useful mechanism can have in the presence of auxiliary information; the obstacle is usefulness itself. Thus, any discussion of disclosure limitation must address auxiliary information. See Section 3 for a successful approach.

2.1 Formalism and the Construction

We begin with a high level description of the result, made formal in the next subsection. Consider any mechanism with the property that it conveys useful information, in the sense that it can be reliably used to determine something about the data set that is not otherwise known to the analyst. Consider also a sensitive predicate of the data, whose value is unknown to the analyst. We construct an auxiliary information generator the couples (XOR's) the useful information provided by the mechanism with the sensitive information. The adversary can then convert the useful information conveyed by the mechanism into an exposure of the sensitive information.

Note that this approach *requires* some notion of *utility* – after all, a mechanism that always outputs the empty string, or a purely random string, clearly preserves privacy[1]. Intuitively, there should be a vector of questions (most of) whose answers *should* be learnable by the analyst, but whose answers are not in general known in advance. We will therefore posit a *utility vector*, denoted with w. We concentrate on the case where this is a binary vector of some fixed length κ[2]. We can think of the utility vector as answers to questions about the (raw) data; we assume w is computable from the raw data. For a sanitization to be useful, a large part of w should be learnable by the analyst.

A *privacy breach* for a database is described by a function (program, Turing machine) $\mathcal{C}$ that takes as input a description of a distribution[3] $\mathcal{D}$ on databases, a database DB drawn according to this distribution, and a string – the purported privacy breach– and outputs a single bit denoting whether the privacy breach occurred or not. We say the adversary *wins*, with respect to $\mathcal{C}$ and for a given $(\mathcal{D}, DB)$ pair, if it produces a string s such that $\mathcal{C}(\mathcal{D}, DB, s)$ accepts. Henceforth "with respect to $\mathcal{C}$" will be implicit. For a privacy breach to meaningful we would require that:

- It should be hard to come up with a privacy breach without getting any information about the database itself, beyond knowledge of the distribution. Otherwise, preserving privacy is trivial - there is none to give up anyway.

[1] The height parable from the Introduction fails in these trivial cases, since it is only through the sanitized version that the adversary learns the average height.

[2] There is nothing special about the use of binary values; one may consider larger domains.

[3] We are agnostic as to *how* a distribution $\mathcal{D}$ is given as input to a function.

- On the other hand, given the actual database it should be possible to find some breach. Otherwise, again, it is very easy to preserve the privacy: just give away the actual data.

To model any additional information the adversary may have about the database we use an auxiliary information generator. This is a function that takes as input a description of the distribution $\mathcal{D}$ from which the database is drawn, as well as the database DB itself, and outputs a string, z, of auxiliary information. This string is given both to the adversary $\mathcal{A}$ and to a *simulator* $\mathcal{A}^*$. The simulator $\mathcal{A}^*$ has no access of any kind to the database, whereas the adversary $\mathcal{A}$ has access to the database via the privacy mechanism.

We model the adversary by an interactive procedure (communicating Turing machine). The theorem below says that for any privacy mechanism San() and any distribution $\mathcal{D}$ satisfying certain technical conditions, there is always some particular piece of auxiliary information, z, so that z alone is useless to someone trying to win, while z in combination with access to the data through the privacy mechanism permits the adversary to win with probability arbitrarily close to 1. In addition to formalizing the entropy requirements on the utility vectors as discussed above, the technical conditions on the distribution say that learning the *length* of a privacy breach does not help one to guess a privacy breach. We state a "meta-theorem" below that captures our results; specific assumptions and the bound Δ are instantiated in Theorems 3 and 4 below.

Theorem 1. [Meta-Theorem.] *For any privacy mechanism* San() *and privacy breach decider* $\mathcal{C}$ *there is an auxiliary information generator* $\mathcal{X}$ *and an adversary* $\mathcal{A}$ *such that for all distributions* $\mathcal{D}$ *such that* $\mathcal{D}$, $\mathcal{C}$ *and* San() *satisfy certain assumptions, and for all adversary simulators* $\mathcal{A}^*$,

$$\Pr[\mathcal{A}(\mathcal{D}, \mathrm{San}(\mathcal{D}, DB), \mathcal{X}(\mathcal{D}, DB)) \text{ wins}] - \Pr[\mathcal{A}^*(\mathcal{D}, \mathcal{X}(\mathcal{D}, DB)) \text{ wins}] \geq \Delta$$

where Δ *is a suitably chosen (large) constant. The probability spaces are over choice of* $DB \in_R \mathcal{D}$ *and the coin flips of* San, $\mathcal{X}$, $\mathcal{A}$, *and* $\mathcal{A}^*$.

The distribution $\mathcal{D}$ completely captures any information that the adversary (and the simulator) has about the database, prior to seeing the output of the auxiliary information generator. For example, it may capture the fact that the rows in the database correspond to people owning at least two pets. Note that in the statement of the meta-theorem all parties have access to $\mathcal{D}$ and may have a description of $\mathcal{C}$ hard-wired in; however, the adversary's strategy does not use either of these.

Approach to Proving Theorem 1. The goal is to begin with a random variable (the utility vector) which is sufficiently hard to guess, and to distill from it a random variable which is close to uniform. This latter random variable can then be used as a one-time pad to create a "secret channel" between the data curator and the data analyst. We define "hard to guess" via *min-entropy*, which measures the probability of the most likely value:

Definition 1. *Let* X *be a discrete random variable obtaining values in* U. *For all* $x \in U$, *let* $p_x = \Pr[X = x]$. *Then the* min-entropy *of* X *is*

$$H_\infty(X) = \min_{x \in U}(-\log p_x) = -(\max_{x \in U} \log p_x) = -\log \max_{x \in U} p_x$$

Thus, we need to convert a random variable with high min-entropy, but which may not be uniform, to one that is close to uniform. The usual tool for this, and one that we employ, is a *randomness extractor*. In particular, we will be using two types of extractors; a *strong extractor* (Definition 2) and a *fuzzy extractor* (Definition 3). The former is used for a slightly weaker result, but achieving better bounds; the latter for the most general result.

2.2 Strategy for $\mathcal{X}$ and $\mathcal{A}$ when all of w is learned from $\mathrm{San}(DB)$

To develop intuition we first describe, slightly informally, the strategy for the special case in which the adversary always learns all of the utility vector, w, from the privacy mechanism. Although this case is covered by the more general one, in which not all of w need be learned, it permits a simpler proof that exactly captures the height example. This is also realistic: the utility vector is completely learned when the sanitization produces a histogram, such as a table of the number of people in the database with given illnesses in each age decile, or a when the sanitizer chooses a random subsample of the rows in the database and reveals the average ages of patients in the subsample exhibiting various types of symptoms. Arguments for this case are presented slightly informally; the more general case has full details.

The distribution $\mathcal{D}$ on databases induces a distribution on the utility vectors.

Assumption 1.

1. For $\ell \in \mathcal{Z}^+$, $\Upsilon \in \mathcal{R}^+$, and $0 \leq \gamma < 1$, *both* the following conditions hold:
   (a) The distribution W on utility vectors has min-entropy at least $3\ell + \Upsilon$.
   (b) Let $DB \in_R \mathcal{D}$. Then with probability at least $1 - \gamma$ over $\mathcal{D}$, DB has a privacy breach of length ℓ; that is, $\Pr[\exists y\, \mathcal{C}(\mathcal{D}, DB, y) = 1] \geq 1 - \gamma$.
2. $\Pr[\mathcal{B}(\mathcal{D}, San(\mathcal{D}, DB)) \text{ wins}] \leq \mu$ for all interactive Turing machines $\mathcal{B}$, where $\mu = \mu(\mathcal{D}, \mathrm{San}())$ is a suitably small constant. The probability is taken over the coin flips of $\mathcal{B}$ and the privacy mechanism San(), as well as the choice of $DB \in_R \mathcal{D}$.

Part (1a) ensures that we can extract ℓ bits of randomness from the utility vector (Claim 2), which can be used as a one-time pad to hide any privacy breach of the same length, and Part (1b) says there is (almost) always some secret of that length to leak. Part 2 is a nontriviality condition saying that the sanitization algorithm provides some privacy, at least against adversaries that have no auxiliary information.

We claim that the assumption implies that conditioned on most privacy breaches of length ℓ, the min-entropy of the utility vector is at least $\ell + \Upsilon$:

Claim 2. *Let $\mathcal{D}$ be as in Assumption 1 Part (1a). Consider any deterministic procedure $\mathcal{P}$ for finding a privacy breach given the database DB, i.e., with high probability over $\mathcal{D}$, $y = \mathcal{P}(DB)$ is a privacy breach. Let $\mathcal{P}(DB) = 0^\ell$ if there is no breach of length ℓ. Consider the distribution Y on y's obtained by sampling $DB \in \mathcal{D}$ and then computing $\mathcal{P}(DB)$. Then with probability at least $1 - 2^{-\ell}$ over $y \in Y$, $H_\infty(W|y) \geq \ell + \Upsilon$.*

Proof. Suppose, for the sake of contradiction, that the event "the conditional min entropy of W drops below $\ell + \Upsilon$, given y", happens with probability $\beta > 2^{-\ell}$ over choice of DB in $\mathcal{D}$. We now argue that the min-entropy of W has to be less than $3\ell + \Upsilon$. Let DB be a database, let w be its utility vector, and let $y = \mathcal{P}(DB)$. Consider the following method for producing $\hat{w}$, a 'guess' for w: choose $\hat{y} \in_R \{0,1\}^\ell$; then, based on $\hat{y}$, take $\hat{w}$ to be the most likely (according to $\mathcal{D}$) value, given $\hat{y}$. The probability that $\hat{y}$ is chosen correctly is $2^{-\ell}$; moreover this even holds conditioned on the event that $H_\infty(W|y) < \ell + \Upsilon$. Given that $\hat{y}$ is correct and the min-entropy of $W|y$ is small, the probability that $\hat{w}$ is correct is greater than $2^{-(\ell+\Upsilon)}$. So, over all, the probability of guessing w is at least $\beta \cdot 2^{-(2\ell+\Upsilon)} > 2^{-(3\ell+\Upsilon)}$, whence the min-entropy of W is strictly less than $3\ell + \Upsilon$. This contradicts Assumption 1a. □

Definition 2. [Strong extractor] *A (k, ε) strong extractor is a function $Ext : \{0,1\}^n \times \{0,1\}^d \mapsto \{0,1\}^m$ such that for every distribution X on $\{0,1\}^n$ with $H_\infty(X) \geq k$ the distribution $U_d \circ Ext(X, U_d)$ is ε-close to the uniform distribution on $\{0,1\}^{m+d}$. The two copies of U_d denote the same random variable.*

The important measure is the entropy loss $k-m$. For us, $n = \kappa$, the length of the utility vector. There are good constructions of strong extractors; see [29, 31]. In particular, from the Leftover Hash Lemma [26] it is known that If $H = \{h : \{0,1\}^n \mapsto \{0,1\}^m\}$ is a pairwise independent family where $m = k - 2\log(1/\varepsilon)$, then $Ext(x, h) \overset{\text{def}}{=} h(x)$ is a strong (k, ε)-extractor. The seed length is $O(n)$, and we can even use logarithmic-sized seeds using almost pairwise independence [23, 32].

Given a (k, ε) strong extractor, with $m \geq \ell$ and $k \geq \ell + \Upsilon$, the strategy for $\mathcal{X}$ and $\mathcal{A}$ is as follows. On input $DB \in_R \mathcal{D}$, $\mathcal{X}$:

- Find a privacy breach y for DB of length ℓ, if one exists, which occurs with probability at least $1-\gamma$ over $\mathcal{D}$[4].
- Compute the utility vector w.
- Choose a seed $s \in_R \{0,1\}^d$ and use the (k, ε) strong extractor to obtain from w an ℓ-bit almost-uniformly distributed string r; that is, $r = \mathrm{Ext}(w, s)$.
- The auxiliary information will be $z = (s, y \oplus r)$.

From the properties of the strong extractor and Claim 2, the distribution on r is within statistical distance $\varepsilon + 2^{-\ell}$ from U_ℓ, the uniform distribution on strings of length ℓ, even given s and y.

Since the adversary learns all of w, from s it can obtain $r = \mathrm{Ext}(s, w)$ and hence the privacy breach y. Since a privacy breach exists with probably at least $1-\gamma$, the adversary $\mathcal{A}$ wins with probability at least $1-\gamma$. We next argue that any adversary simulator $\mathcal{A}^*$ wins with probability bounded above by $\mu + \varepsilon + 2^{-\ell}$, yielding a gap of at least $1 - (\gamma + \mu + \varepsilon + 2^{-\ell})$.

Assumption 1(2) implies that $\Pr[\mathcal{A}^*(\mathcal{D}) \text{ wins}] \leq \mu$. Let ρ denote the maximum, over all $y \in \{0,1\}^\ell$, of the probability, over choice of $DB \in_R \mathcal{D}$, that y is a privacy breach for DB. Assumption 1(2) also implies that $\rho \leq \mu$.

By Assumption 1(1a), even conditioned on y, the extracted r is (almost) uniformly distributed, and hence so is $y \oplus r$. Consequently, $\forall z$, the probability that $\mathcal{X}$ produces z is essentially independent of y. Thus, the simulator's probability of producing a privacy breach of length ℓ for the given database is bounded by $\rho + \varepsilon + 2^{-\ell} \leq \mu + \varepsilon + 2^{-\ell}$, as it can generate simulated "auxiliary information" with a distribution within distance ε of the correct one. We conclude:

Theorem 3. [Full Recovery of Utility Vector] *Let Ext be a (k, ε) strong extractor. For any privacy mechanism* San() *and privacy breach decider $\mathcal{C}$ there is an auxiliary information generator $\mathcal{X}$ and an adversary $\mathcal{A}$ such that for all distributions $\mathcal{D}$ such that $\mathcal{D}$, $\mathcal{C}$ and* San() *satisfy Assumption 1 with $k \geq \ell + \Upsilon$ and for all adversary simulators $\mathcal{A}^*$,*

$$\Pr[\mathcal{A}(\mathcal{D}, \mathrm{San}(\mathcal{D}, DB), \mathcal{X}(\mathcal{D}, DB)) \text{ wins}] - \Pr[\mathcal{A}^*(\mathcal{D}, \mathcal{X}(\mathcal{D}, DB)) \text{ wins}] \geq \Delta$$

where $\Delta = 1 - (\gamma + \mu + \varepsilon + 2^{-\ell})$. The probability spaces are over choice of $DB \in_R \mathcal{D}$ and the coin flips of San, *$\mathcal{X}$, $\mathcal{A}$, and $\mathcal{A}^*$.*

Using the constructions of strong extractors based on pairwise independence it suffices that $\Upsilon \geq 2\log 1/\varepsilon$.

[4] We will discuss efficiency in Section 2.3.1.

2.3 Strategy when w need not be fully learned:

The more interesting case is when the sanitization does not necessarily reveal all of w; rather, the guarantee is only that it always reveal a vector w' within Hamming distance κ/c of w for constant c to be determined[5]. The difficulty with the previous approach is that if the privacy mechanism is randomized, then the auxiliary information generator may not know which w' is seen by the adversary. Thus, even given the seed s, the adversary may not be able to extract the same random pad from w' that the auxiliary information generator extracted from w (as is the case, for example, in the extractors based on pair-wise independent hash functions). *This* problem is solved by using *fuzzy extractors* [13].

Definition 3. *An* $(\mathcal{M}, m, \ell, k', t, \varepsilon)$ fuzzy extractor *is given by two procedures* (Gen,Rec)*, for generating and reconstructing.*

1. Gen(w) *is a randomized generation procedure. On input* $w \in \mathcal{M}$ *outputs an "extracted" string* $r \in \{0,1\}^\ell$ *and a public string* p*. For any distribution* W *on* $\mathcal{M}$ *of min-entropy* m*, if* $(R, P) \leftarrow \mathrm{Gen}(W)$ *then the distributions* (R, P) *and* (U_ℓ, P) *are within statistical distance* ε*.*

2. Rec(p, w') *is a deterministic reconstruction procedure allowing recovery of* $r = R(w)$ *from the corresponding public string* $p = P(w)$ *together with any vector* w' *of distance at most* t *from* w*. That is, if* $(r, p) \leftarrow \mathrm{Gen}(w)$ *and* $||w - w'||_1 \leq t$ *then* $\mathrm{Rec}(w', p) = r$*.*

In other words, $r = R(w)$ looks uniform, even given $p = P(w)$, and $r = R(w)$ can be reconstructed from $p = P(w)$ and any w' sufficiently close to w. There exist efficient fuzzy extractors where the two procedures run in time polynomial in the representation size of an element in $\mathcal{M}$ [13, 14]. The important parameters for us are the entropy loss $(m - \ell)$ and the length of the public string (k').

Constructions of fuzzy extractors can be based on error-correcting codes. Roughly speaking, let ECC be a code where any two codewords differ by at least $2t$ bits. The generation procedure $Gen(w)$ is: choose r' uniformly at random; let r be extracted from r', and let $p = w \oplus ECC(r)$. Note that if w' is close to w, then $w' \oplus p$ is close to $ECC(r')$. Given p and w': compute $w' \oplus p$ and then apply the decoding procedure of ECC to retrieve r', and then extract r. See[13].

Assume a sanitization mechanism San() and a distribution $\mathcal{D}$ on databases, where $\mathrm{San}(\mathcal{D}, DB)$ always reveals a vector within Hamming distance κ/c of the real utility vector w. The sanitization mechanism implicitly defines a distribution on these revealed vectors, denoted $\mathrm{Utility}_{\mathrm{San}}(\mathcal{D})$.

We now strengthen Part (1a) of Assumption 1 to say that the entropy of the source $\mathrm{Utility}_{\mathrm{San}}(\mathcal{D})$ is sufficiently high, so that even conditioned on privacy breach y of length ℓ **and** $P = \mathrm{Gen}(W)$ there is sufficient entropy to encode y.

Assumption 2. Same as Assumption 1, but with(1a) replaced with: the min-entropy of $\mathrm{Utility}_{\mathrm{San}}(\mathcal{D})$ is at least $k' + 3\ell$, where k' is the length of the public string p produced by the fuzzy extractor.

By Claim 2, this means that with probability at least $1 - 2^{-\ell}$ over $y \in Y$, $H_\infty(\mathrm{Utility}_{\mathrm{San}}(\mathcal{D})|y) \geq \ell + k'$.

Repeating the strategy of $\mathcal{X}$ from Section 2.2, while substituting the strong extractor with a fuzzy extractor raises a new problem: the public string p might leak information about w: not

[5] One could also consider privacy mechanisms that produce good approximations to the utility vector with a certain probability for the distribution $\mathcal{D}$, where the probability is taken over the choice of $DB \in_R \mathcal{D}$ and the coins of the privacy mechanism. The theorem and proof hold *mutatis mutandis.*

enough to learn w, but enough to disclose *some* privacy breach y. To overcome this possibility we will use the fact that the presumed sanitization procedure does not leak a privacy breach.

For a given database DB, let w be the utility vector. This can be computed by $\mathcal{X}$, who has access to the database. $\mathcal{X}$ simulates interaction with the privacy mechanism to obtain a non-disclosive w' close to w (within Hamming distance κ/c). The auxiliary information generator runs $\mathrm{Gen}(w')$, obtaining $(r = R(w'), p = P(w'))$. It chooses a privacy breach y of length ℓ, assuming one exists. It then sets $z = (p, r \oplus y)$.

Let w'' be the version of w seen by the adversary. Clearly, assuming $2\kappa/c \leq t$ in Definition 3, the adversary can reconstruct r. This is true since w' and w'' are both within distance κ/c of w, and hence they are within distance $2\kappa/c$ of each other. So w'' is within the "reconstruction radius" for any $r \leftarrow \mathrm{Gen}(w')$. Once the adversary has reconstructed r, obtaining y is immediate. Thus the adversary is able to produce a privacy breach with probability at least $1 - \gamma$.

It remains to analyze the probability with which the simulator, having access only to z but not to the privacy mechanism (and hence, not to any w'' close to w), produces a privacy breach.

In the sequel, let $\mathcal{B}$ always denote the *best* machine, among all those with access to the given information, at producing producing a privacy breach ("winning").

By Assumption 1(2), $\Pr[\mathcal{B}(\mathcal{D}, \mathrm{San}(\mathcal{D}, DB)) \text{ wins}] \leq \mu$, where the probability is taken over the coin tosses of the privacy mechanism and the machine $\mathcal{B}$, and the choice of $DB \in_R \mathcal{D}$. Since $p = P(w')$ is computed from w', which in turn is computable from $\mathrm{San}(DB)$, we have

$$p_1 = \Pr[\mathcal{B}(\mathcal{D}, p) \text{ wins}] \leq \mu$$

where the probability space is now also over the choices made by Gen(), that is, the choice of $p = P(w')$. Now, let U_ℓ denote the uniform distribution on ℓ-bit strings. Concatenating a random string $u \in_R U_\ell$ to p cannot help $\mathcal{B}$ to win, so

$$p_2 = \Pr[\mathcal{B}(\mathcal{D}, p, u) \text{ wins}] = p_1 \leq \mu$$

where the probability space is now also over choice of u. For any fixed string $y \in \{0,1\}^\ell$ we have $U_\ell = U_\ell \oplus y$, so for all $y \in \{0,1\}^\ell$, and in particular, for all privacy breaches y of DB,

$$p_3 = \Pr[\mathcal{B}(\mathcal{D}, p, u \oplus y) \text{ wins}] = p_2 \leq \mu.$$

From Assumption 2 we know that for a DB chosen according to $\mathcal{D}$, we have that given $y = \mathcal{P}(DB)$ the distributions $(P, R) = \mathrm{Gen}(W')$, and (P, U_ℓ) have distance at most $\varepsilon + 2^{-\ell}$. It follows that

$$p_4 = \Pr[\mathcal{B}(\mathcal{D}, p, r \oplus y) \text{ wins}] \leq p_3 + \varepsilon + 2^{-\ell} \leq \mu + \varepsilon + 2^{-\ell}.$$

Now, p_4 is an upper bound on the probability that the *simulator* $\mathcal{A}^*$ wins, given $\mathcal{D}$ and the auxiliary information $z = (p, r \oplus y)$, so

$$\Pr[\mathcal{A}^*(\mathcal{D}, z) \text{ wins}] \leq p_4 \leq \mu + \varepsilon + 2^{-\ell}.$$

Thus the gap between the winning probabilities of the adversary and the simulator is at least

$$(1 - \gamma) - (\mu + \varepsilon + 2^{-\ell}) = 1 - (\gamma + \mu + \varepsilon + 2^{-\ell}).$$

Setting $\Delta = 1 - (\gamma + \mu + \varepsilon + 2^{-\ell})$ proves the following theorem.

Theorem 4. [Partial Recovery of Utility Vector] *Let* FExt *be an* $(\mathcal{M}, m, \ell, k', t, \varepsilon)$ *fuzzy extractor. For any privacy mechanism* San() *and privacy breach decider* $\mathcal{C}$ *there is an auxiliary information generator* $\mathcal{X}$ *and an adversary* $\mathcal{A}$ *such that for all distributions* $\mathcal{D}$ *for which:*

1. *San(DB) produces a vector that is within distance κ/c of w, where $2\kappa/c \leq t$; and*

2. *$\mathcal{D}$, $\mathcal{C}$ and* San() *satisfy Assumption 2 where ℓ and k' are as in* FExt*,*

and for all adversary simulators $\mathcal{A}^*$:

$$\Pr[\mathcal{A}(\mathcal{D}, \mathrm{San}(\mathcal{D}, DB), \mathcal{X}(\mathcal{D}, DB)) \text{ wins}] - \Pr[\mathcal{A}^*(\mathcal{D}, \mathcal{X}(\mathcal{D}, DB)) \text{ wins}] \geq \Delta$$

where $\Delta = 1 - (\gamma + \mu + \varepsilon + 2^{-\ell})$. The probability spaces are over choice of $DB \in_R \mathcal{D}$ and the coin flips of San*, $\mathcal{X}$, $\mathcal{A}$, and $\mathcal{A}^*$.*

2.3.1 Comments

Fuzzy Extractors: A good fuzzy extractor "wastes" little of the entropy on the public string. Better fuzzy extractors are better for the adversary, since the attack requires ℓ bits of residual min-entropy after the public string has been generated.

Unlike most applications of fuzzy extractors (see, in particular, [13, 14]), in this proof we are not interested in hiding partial information about the source, in our case the approximate utility vectors w' revealed by the San mechanism, so we don't care how much min-entropy is used up in generating p. We only require sufficient residual min-entropy for the generation of the random pad r. This is because information revealed by the privacy mechanism is not itself disclosive; indeed it is by definition safe to release. Similarly, we don't necessarily need to maximize the tolerance t, although if we have a richer class of fuzzy extractors the impossibility result applies to more relaxed privacy mechanisms (those that reveal worse approximations to the true utility vector).

Other distance measures: we have concentrated on the Hamming distance of the the noisy utility vector w' from the original one. There are fuzzy extractors of other distance measures, such as the edit distance or set difference as well as general metric spaces that will fit nicely when the utility vector retrieved is close to the original in those distance measures - see [13] (full version).

Complexity: One possible recourse from impossibility results is arguing that performing the attack is too costly computationally wise and hence not likely to occur. However, this is not the case here: provided finding a privacy breach y, given DB, can be done efficiently, the complexity of $\mathcal{X}$ and $\mathcal{A}$ are both low and simply requires simulating the sanitizing process and the legitimate database user (who must be efficient for the system to be useful).

3 The Case for Differential Privacy

A surprising aspect of the height example is that the compromise can occur even if Terry Gross is not in the database. In order to sidestep this issue, Dwork and McSherry shift from absolute guarantees about disclosures to relative ones: any given disclosure – indeed, any output at all of the privacy mechanism – should be, within a small multiplicative factor, just as likely independent of whether any individual opts in to, or opts out of, the database. As a consequence, there is a nominally increased risk to the individual in participating, and only nominal gain to be had by concealing or misrepresenting one's data. Note that a bad disclosure can still occur, but the differential privacy guarantee assures the individual that it will not be the presence of her data that causes it, nor could the disclosure be avoided through any action or inaction on the part of the user.

Two data sets differ on at most one element if one is a subset of the other and the larger contains at most one additional element.

Definition 4. [Dwork and McSherry, 2006] *A randomized function $\mathcal{K}$ gives* ϵ-differential privacy *if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$,*

$$\Pr[\mathcal{K}(D_1) \in S] \quad \leq \quad \exp(\epsilon) \times \Pr[\mathcal{K}(D_2) \in S] \tag{1}$$

A mechanism $\mathcal{K}$ satisfying this definition addresses concerns that any participant might have about the leakage of her personal information x: even if the participant removed her data from the data set, no outputs (and thus consequences of outputs) would become significantly more or less likely. For example, if the database were to be consulted by an insurance provider before deciding whether or not to insure Terry Gross, then the presence or absence of Terry Gross in the database will not significantly affect her chance of receiving coverage.

This definition extends to group privacy as well. A collection of c participants might be concerned that their collective data might leak information, even when a single participant's does not. Using this definition, we can bound the dilation of any probability by at most $\exp(\varepsilon c)$, which may be tolerable for small c. Note that we specifically aim to disclose aggregate information about large groups, so we should expect privacy bounds to disintegrate with increasing group size.

A general technique for achieving ϵ-differential privacy for any ϵ appears in [16]. See also [30, 28, 5, 4, 30, 3, 27] for other works dealing with the notion.

4 Discussion

It is initially puzzling that Dalenius' goal, formalized as a relaxed version of semantic security, cannot be achieved, while semantic security for cryptosystems can be achieved (under reasonable computational assumptions). This discrepancy arises from the utility requirement. Our adversary is analogous to the eavesdropper, while our user is analogous to the message recipient, and yet there is no decryption key to set them apart, they are one and the same. Very roughly, the database is designed to convey certain information. An auxiliary information generator knowing the data therefore knows much about what the user will learn from the database. This can be used to establish a shared secret with the adversary/user that is unavailable to anyone not having access to the database.

A natural question is the relationship between code obfuscation and statistical databases. The goal in code obfuscation is to be able to publish a program "while hiding its guts" - intuitively users should learn no more than what they would learn with black-box (oracle) access to the program. A basic impossibility result of Barak *et al.* [6] says that some functions cannot be obfuscated.

Obfuscation and sanitization might appear to be related as follows: One can think of the database as the program to be obfuscated, and the queries whose answers are the bits of the utility vector as the inputs to the program. This intriguing, albeit superficial, similarity suggests further study of the connection.

Acknowledgements

The authors gratefully acknowledge many helpful discussions with Prahladh Harsha. Thanks also go to Shuchi Chawla, Amos Fiat, Frank McSherry, Ilya Mironov, Kobbi Nissim, Omer Reingold, Adam Smith, Madhu Sudan, and Hoeteck Wee. Our interest in privacy was inspired by conversations with Helen Nissenbaum.

References

[1] N. R. Adam and J. C. Wortmann, Security-Control Methods for Statistical Databases: A Comparative Study, *ACM Computing Surveys* 21(4): 515-556 (1989).

[2] R. Agrawal and R. Srikant, Privacy-preserving data mining, *Proc. ACM SIGMOD International Conference on Management of Data*, pp. 439–450, 2000.

[3] A. Blum, K. Ligett and A. Roth, *A learning theory approach to non-interactive database privacy*, Proceedings of the 40th ACM Symposium on Theory of Computing, 2008, pp. 609–618.

[4] A. Blum, C. Dwork, F. McSherry, and K. Nissim, *Practical privacy: The SuLQ framework*, Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2005, pp. 128–138.

[5] B. Barak, K. Chaudhuri, C. Dwork, S. Kale, F. McSherry and K. Talwar, *Privacy, accuracy, and consistency too: a holistic solution to contingency table release*, Proceedings of the 24th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2007, pp. 273–282.

[6] B. Barak, O. Goldreich, R. Impagliazzo, S. Rudich, A. Sahai, S. P. Vadhan, K. Yang, *On the (Im)possibility of Obfuscating Programs*, Proceedings of CRYPTO 2001, pp. 1–18.

[7] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee. Toward privacy in public databases. In *Proceedings of the 2nd Theory of Cryptography Conference*, pages 363–385, 2005.

[8] S. Chawla, C. Dwork, F. McSherry and K. Talwar, *Privacy-preserving histograms*, Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence, 2005.

[9] T. Dalenius, Towards a methodology for statistical disclosure control. *Statistik Tidskrift 15*, pp. 429–444, 1977.

[10] D. E. Denning, *Secure statistical databases with random sample queries*, ACM Transactions on Database Systems, 5(3):291–315, September 1980.

[11] I. Dinur and K. Nissim. *Revealing information while preserving privacy*, Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, 2003, pp. 202–210.

[12] D. Dobkin, A.K. Jones, and R.J. Lipton, *Secure databases: Protection against user influence*, ACM Trans. Database Systems, 4(1), pp. 97–106, 1979.

[13] Y. Dodis, L. Reyzin and A. Smith, *Fuzzy extractors: How to generate strong keys from biometrics and other noisy data*, Proceedings of EUROCRYPT 2004, pp. 523–540, 2004.

[14] Y. Dodis and A. Smith, *Correcting Errors Without Leaking Partial Information*, Proceedings of the 37th ACM Symposium on Theory of Computing, pp. 654–663, 2005.

[15] C. Dwork and F. McSherry, personal communication, 2006.

[16] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd Theory of Cryptography Conference*, pages 265–284, 2006.

[17] C. Dwork and K. Nissim, *Privacy-preserving datamining on vertically partitioned databases*, Advances in Cryptology: Proceedings of Crypto 2004, pp. 528–544, 2004.

[18] A. Evfimievski, J. Gehrke, and R. Srikant, *Limiting privacy breaches in privacy preserving data mining*, Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, pp. 211–222, June 2003.

[19] O. Goldreich, **Modern Cryptography, Probabilistic Proofs and Pseudorandomness**, Springer, Algorithms and Combinatorics, Vol 17, 1998.

[20] O. Goldreich, **Foundations of Cryptography: Basic Tools**, Cambridge U. Press, 2001.

[21] O. Goldreich, S. Micali and A. Wigderson, *How to Play any Mental Game, or A Completeness Theorem for Protocols with Honest Majority*, Proceedings of STOC 1987, pp. 218–229.

[22] O. Goldreich,, Y. Oren, *Definitions and Properties of Zero-Knowledge Proof Systems*, J. Cryptology 7(1), 1994, pp. 1–32.

[23] O. Goldreich and A. Wigderson, *Tiny families of functions with random properties: A quality-size trade-off for hashing*, Random Structures and Algorithms 11(4), 1997, pp. 315–343.

[24] S. Goldwasser and S. Micali, *Probabilistic encryption*, Journal of Computer and System Sciences 28, pp. 270–299, 1984; prelminary version appeared in *Proceedings 14th Annual ACM Symposium on Theory of Computing*, 1982.

[25] S. Goldwasser, S. Micali and R. L. Rivest, *A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks*, SIAM J. Computing 17(2), 1988, pp. 281–308.

[26] R. Impagliazzo, L. Levin and M. Luby, *Pseudo-random Generation from one-way functions*, Proc 21st ACM Symposium on the Theory of Computing, 1989, pp. 12–24

[27] S. Kasiviswanthan, H. Lee, K. Nissim, S. Raskhodnikova and A. Smith, *What Can Be Learned Privately?*, to appear, FOCS 2008.

[28] F. McSherry and K. Talwar, *Mechanism Design via Differential Privacy*, FOCS 2007, pp. 94–103.

[29] N. Nisan and D. Zuckerman. *Randomness is linear in space*, J. Comput. Syst. Sci., 52(1):43–52, 1996.

[30] K. Nissim, S. Raskhodnikova and A. Smith, *Smooth sensitivity and sampling in private data analysis*, STOC 2007, pp. 75–84.

[31] R. Shaltiel, *Recent developments in explicit constructions of extractors*, Bulletin of the EATCS, 77, 2002, pp. 67–95.

[32] A. Srinivasan and D. Zuckerman, *Computing with Very Weak Random Sources*, SIAM J. Comput. 28(4): 1433-1459 (1999) (Preliminary version in FOCS 1994).

[33] L. Sweeney, *Weaving technology and policy together to maintain confidentiality*, J. Law Med Ethics, 1997, 25(2-3), pp. 98–110.

[34] L. Sweeney, Achieving k-anonymity privacy protection using generalization and suppression. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 10* (5), 2002; 571-588.