1 Motivation

Protect information contained in statistical database, i.e, if a database is a representative sample of an underlying, the goal of a privacy preserving statistical database is to enable the user to learn properties of the population as a whole , while protecting the privacy of the individuals in the sample.
What we want to achieve is something close to semantic security. Dalenius in 1977 articulated a desired property for database: nothing abouth an individual should be learnable from a database that cannot be learned without access to the database. This kind of privacy cannot be achieved: the obstacle is *auxiliary information*. Here an example:
suppose one's exact height were considered sensitive information and revealing it were a privacy breach. Assume that the database yields the average heights of women of different nationalities. AN adversary who access the 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 database ,learns relatively little.
There are to remarks to do wrt the impossibility results:

- it applies regardless of whether or not Terry Gross is in the database
- Dalenius goal cannot be achieved while semantic security for crypto yes.

The first of these leads naturally to a new approach to formulating privacy goals: the risk to ones privacy, or in general, any type of risk, such as the risk of being denied automobile insurance, should not substantially increase as a result of participating in a statistical database. This is captured by *differential privacy*

2 Differential Privacy: the setting

Given the impossibility result just showed we need to relax the definition of privacy: we move from absolute guarantees about disclosures to relative ones; any given disclosure will be, within a small multiplicative factor.

Definition 2.1. A randomized algrotihm A 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(A),

$$Pr[M(D_1) \in S] \leq exp(\epsilon) \times Pr[A(M_2) \in S] \tag{1}$$

If $D_1 - D_2 > 1$ the equation becomes:

$$Pr[M(D_1) \in S] \leq exp(\epsilon \times (D_1 \oplus D_2)) \times Pr[A(M_2) \in S] \tag{2}$$

Where $\oplus$ represents the symmetric difference between the two datasets.
A mechanism M 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.

3 Achieving DP

Now that we defined what we want to achieve, let's describe a concrete mechanism achieving differential privacy.
The mechanism works by adding appropriately chosen **random noise** (form now on when we talk about random nois ewe consider Laplace random noise) to the answer a = f(X), where f is the query function and X is the database; thus the query functions may operate on the entire database at once. It can be as simple as eg, "Count the number of rows in the database satisfying a given predicate". Obviously there are also more complicated example that we will see later.

We will achieve ϵ-differential privacy by the addition of random noise whose magnitude is chosen as a function of the largest change a single participant could have on the output to the query function; we refer to this quantity as the **sensitivity (or stability as it is called in ref)** of the function.

Definition 3.1. For $f : D \to R^d$ the sensitivity of f is:

$$\Delta f = max_{D_1,D_2} ||f(D_1) - f(D_2)|| \tag{3}$$

for all database D_1, D_2 differing in at most one element

For many types of queries Δf will be quite small. In particular, the simple counting queries ("How many rows have property P?") have $\Delta f \leq 1$. Our techniques work best ie, introduce the least noise when Δf is small. **Note that sensitivity is a property of the function alone, and is independent of the database**.

4 Aggregation

The simplest differentially-private aggregation releases the number of records in a data set, perturbed by symmetric exponential (Laplace) noise, with density function $p(x) \propto exp(|x|)$. In fact adding symmetric exponential noise to counts causes the probability of any output (or set of outputs) to increase or decrease by at most a multiplicative factor when the counts are translated. Changing an input data set from A to B can shift the true count by at most $|A \oplus B|$. The Laplace distribution is chosen because it has the property that translating its

center (shifting the true value) by one unit scales the probability of any output by a multiplicative factor of at most exp(1). If the noise is first multiplied by $\frac{1}{\epsilon}$ this becomes $exp(\epsilon)$, resulting in ϵ-differential privacy.

Theorem 4.1. The mechanism $M(X) = |X| + \ Laplace(\frac{1}{\epsilon})$

The theorem works for the count transformation that is "1-stable" or in other word which sensitivity is 1. Let's see how the definition changes for a generic transformation.

Definition 4.2. We say that a transformation T is c-stable if for any two input dataset A,B:

$$T(A) \oplus T(B) \leq c \times |A \oplus B| \tag{4}$$

and

Theorem 4.3. Let M provide ϵ-DP, and let T be a an arbitrary c-stable transformation. The composite computation $M \circ T$ provides$(\epsilon \times c)$-differential privacy.

5 Composition

5.1 Sequential composition

Any sequence of computations that each provide differential privacy in isolation also provide differential privacy in sequence. Importantly, this is true not only when they are run independently, but even when subsequent computations can incorporate the outcomes of the preceding computations. Notationally, we explicitly index each computation by the preceding outcomes, allowing them to vary arbitrarily as a function of these values. We still require each computation satisfy differential privacy with respect to their input data.

Theorem 5.1. Let Mi each provide ϵ-differential privacy. The sequence of $M_i(X)$ provides $\sum_i$-differential privacy

5.2 Parallel composition

While general sequences of queries accumulate privacy costs additively, when the queries are applied to disjoint subsets of the data we can improve the bound. Specifically, if the domain of input records is partitioned into disjoint sets, independent of the actual data, and the restrictions of the input data to each part are subjected to differentially-private analysis, the ultimate privacy guarantee depends only on the worst of the guarantees of each analysis, not the sum.

Theorem 5.2. Let M_i each provide ϵ-differential privacy. Let D_i be arbitrary disjoint subsets of the input domain D. The sequence of $Mi(X \bigcap D_i$ provides ϵ-differential privacy