Decision Support for Sharing Data Using Differential Privacy

Mark F. St. John* Pacific Science & Engineering
Grit Denker† SRI International
Peeter Laud‡ Cybernetica AS
Karsten Martiny§ SRI International
Alisa Pankova¶ Cybernetica AS
Dusko Pavlovic‖ University of Hawaii

ABSTRACT

Owners of data may wish to share some statistics with others, but they may be worried of privacy of the underlying data. An effective solution to this problem is to employ provable privacy techniques, such as differential privacy, to add noise to the statistics before releasing them. This protection lowers the risk of sharing sensitive data with more or less trusted data sharing partners. Unfortunately, applying differential privacy in its mathematical form requires one to fix certain numeric parameters, which involves subtle computations and expert knowledge that the data owners may lack.
In this paper, we first describe a differential privacy parameter selection procedure that minimizes what lay data owners need to know. Second, we describe a user visualization and workflow that makes this procedure available for lay data owners by helping them set the level of noise appropriately to achieve a tolerable risk level. Finally, we describe a user study in which human factors professionals who were naïve to differential privacy were briefly trained on the concept of using differential privacy for data sharing and then used the visualization to determine an appropriate level of noise.

1 INTRODUCTION

Owners of data may wish to share statistical data with others, but they may not want to reveal the underlying data, which may be sensitive. Rather, the data owner desires to keep the underlying data private.

For example, during an epidemic, a government might wish to share counts of disease state per village with various international health organizations. However, the state of individuals is very sensitive, and the government wants to keep these data private. Unfortunately, even if only aggregated data is released, there is still a considerable chance that a seemingly anonymized dataset may reveal the states of specific individuals. Even worse, information about specific individuals can potentially be correlated with other datasets, which might even be released in the future, causing the released data to become increasingly vulnerable to re-identification.

In some cases, if privacy cannot be guaranteed, a data owner may choose to forego data sharing all together. This unfortunate situation leads to operational risks or lost opportunities that sharing the data might have produced. How can we facilitate data sharing, while ensuring that the underlying data remain private?

An effective solution to this problem is to employ provable privacy techniques before releasing the data to ensure that privacy is sufficiently protected, even if additional datasets are released in the future [20]. Differential privacy [9] is one such technique that can be used to protect shared statistical data by adding noise to the shared statistics. It is designed with parameters that impose upper bounds on the likelihood of guessing the underlying data by adding specific amounts of noise to the statistics. The added noise provides a guarantee that the likelihood of guessing can be kept to a desired low level.

In turn, the guessing probability impacts the risk of sharing data. Data sharing risk is a combination of the sensitivity of the data, the trust in the data sharing partner, and the guessing probability. Data sensitivity is an assessment of the damage that could occur if the data were exploited. Partner trust is an assessment of how likely that data sharing partner is to exploit the data. For sharing a given set of data with a given partner, the data owner needs to determine a noise level and probability of guessing that leads to a tolerable level of risk.

Adding noise to the shared statistics, makes the data potentially less useful. The data owner must determine a level of noise that is suitably high to keep the underlying data from being guessed but suitably low to keep the data useful. Differential privacy can help with this problem by calculating the guessing probability for a given data set and then computing the required amount of noise to add to the shared data to guarantee a desired low probability of guessing. Data owners must then assess, and perhaps modify, the noise and the guessing probability to achieve a balance between risk and utility. Unfortunately, applying differential privacy in its mathematical form requires one to fix certain numeric parameters, which involves subtle computations and expert knowledge. This constraint has been noted as a significant practical obstacle to differential privacys application, as data owners often lack the necessary expertise to apply differential privacy on their own.

Here, we describe a differential privacy parameter selection procedure for lay data owners. The main component of the procedure is a visualization mechanism that depicts the consequences of making a certain choice of parameters in terms that the data owner should be able to understand. The data owner will be able to explore the space of parameters and set the level of noise appropriately to achieve a tolerable risk level, while, at the same time, give useful information for the data sharing partner. This solution will enable data owners to share statistical data, while remaining confident that the underlying data remains private.

Hence, our set-up is the following. There is a data owner that holds a database. He understands the collected data sufficiently well to figure out its *sensitivity*[1], i.e. the amount of damage caused by a breach. There is an analyst that makes a query against this database. The owner answers this query, but with the help of a differential privacy mechanism. The owner knows the analyst well, and can judge his trustworthyness. Importantly, the data owner also knows, how the usefulness of the answer of the query to the analyst is affected by the amount of noise added to the answer. Our visualization mechanism allows the data owner to compare the amount of added noise with the risk of data sharing. The computations behind the

*e-mail: markst.john@pacific-science.com
†e-mail: grit.denker@sri.com
‡e-mail: peeter.laud@cyber.ee
§e-mail: karsten.martiny@sri.com
¶e-mail: alisa.pankova@cyber.ee
‖e-mail: dusko@dusko.org

[1]In differential privacy literature, *sensitivity* is a property of functions between metric spaces. This notion will not play a significant role in this paper. Our use of "sensitivity" is common in certain communities where the sharing of data is a long-standing issue.

2639-4332/21/$31.00 ©2021 IEEE
DOI 10.1109/VizSec53666.2021.00008

visualization depend on the database schema and on the query, as well as on the way of adding and measuring the noise, but the visualization itself, including the manner in which the analyst interacts with it stays basically the same for different datasets and queries.

We first provide an overview of differential privacy and a mechanism that is appropriately designed as a back end for the visualization. Then we describe a visualization to help data owners explore the relationships between data sensitivity, trust, noise, and risk and set an appropriate level of noise to achieve a tolerable level of risk and utility. In particular, users will be able to modify the amount of added noise and view the impact on guessing probability and data sharing risk. Finally, we describe a user study in which human factors professionals who were naïve to differential privacy were briefly trained on the concept of using differential privacy for data sharing and then used the visualization to determine an appropriate level of noise. Feedback from the participants was used to develop a revised visualization design.

It takes several steps to arrive at the mechanism suitable for visualization, and also matching our expectations of privacy and utility properties. These steps switch back and forth between considering achievable privacy and utility properties, and their connections to differential privacy properties and mechanisms. The chosen mechanisms are related to the desired privacy and utility properties, and different desiderata may lead to different mechanisms (still, the visualization and the data owner's workflow stays the same), but our privacy properties are standard, and utility properties likewise, hence the differential privacy mechanism that we use is also one that could be expected to turn up in many scenarios. These steps, which should be considered as a separate contribution of this paper, are given in Sec. 2.

Running Example. In our example database, there is a table about residents of some country, containing the attributes *ID* and *disease state*. A disease state is represented by an ENUM datatype which can take one of the four values: *S* (susceptible), *I* (infected), *R* (recovered), *D* (deceased). There is also a list of villages, and an assignment of residents to villages. This assignment is assumed to be public. The communications officer desires to share counts of disease state per village. Characteristics of the data set, including its size and sparseness, determine how much the attacker can learn about underlying data from the observed counts. Our running example is specific, and is based on discrete inputs, but the considerations that lead to the design choices of the DP parameter selection tool are general.

2 Differential privacy and data sharing risk

Differential privacy (DP) [9] is used to quantitatively define privacy losses coming from answers to statistical queries about data collections. Roughly speaking, if two databases are sufficiently similar, then the attacker should not be able (up to a certain extent, defined by a special privacy parameter ε) to distinguish between them after observing the query output.

Definition 2.1 *(Differential Privacy [9]) Let X be the set of all possible databases to which a query may be applied. Let $\varepsilon \geq 0$. A mechanism M is* ε-differentially private *if, for any two database instances $x_0, x_1 \in X$, and for any subset $Y \subseteq M(X)$ of outputs, we have* $\Pr[M(x_0) \in Y] \leq e^{\varepsilon \cdot d(x_0,x_1)} \cdot \Pr[M(x_1) \in Y]$.

The distance $d(x_0,x_1)$ in Definition 2.1 can be defined in different ways. For example, it can be the number of different rows in two tables. In our running example of an SIRD count histogram, we care about the disease state of residents, so we can define $d(x_0,x_1) = k$ iff there are exactly k residents in x_0, whose disease states are different from the states that they have in x_1.

A general method for making an information release mechanism with numeric output ε-differentially private is to add random noise of appropriate magnitude to the output of $M(x)$. This magnitude depends not only on ε, but also on the query sensitivity of the mechanism, i.e., the amount of change of its output when its input is changed by a unit amount. For example, ε-DP can be achieved applying Laplace mechanism (described e.g., in [10]), which samples noise from Laplace distribution $\mathrm{Lap}(\frac{\Delta f}{\varepsilon})(x) \sim \frac{\varepsilon}{2\Delta f} \cdot e^{-\frac{\varepsilon|x|}{\Delta f}}$, where Δf is the query sensitivity.

The more noise we add, the more private the data release becomes, but at the same time, the utility of the data released in this way may decrease. The question is how to find the ε for which privacy and utility are balanced.

Consider the running example of an SIRD count histogram. It may seem that releasing just an aggregated statistic is safe if the analyst, the data sharing partner, only observes the aggregated counts. However, this is not always the case. If the analyst already knows the disease state of many residents in the dataset (e.g., if the data table represents a small village), then it will be easier for the analyst to guess which disease state a particular person in this dataset may have. In an extreme case, if the analyst knows in advance that there are m people already infected, the total count is $m+1$, and the only person whose disease status is so far unknown is Alice, then the count histogram will release the disease state of Alice completely. The goal of differential privacy is to protect even against such knowledgeable attackers.

2.1 Estimating the risk

Intuitively, differential privacy quantifies how much the distribution of query outputs changes if the disease state of some individual has changed, and smaller ε means more similarity. Hence, it is closely related to the ability of an analyst to guess the disease state of a particular user. However, the definition of DP allows ε to be arbitrarily large, and does not give enough intuition concerning how small ε would give enough privacy. Therefore, we want to convert this definition to a more intuitive leakage metric.

A useful construct is data sharing risk. Across multiple industries and fields of study, risk is defined as

$$\textit{risk} = \textit{value of asset} \cdot \textit{chance of loss} \quad . \tag{1}$$

In the context of data sharing, the value of the asset is the sensitivity of the data and the degree to which the exploitation of that data could harm the organization that shared the data. The chance of loss is based on the (dis)trust in the data sharing partner. That is, the chance of loss is low for a trusted data sharing partner, but the chance of loss is high for a distrusted data sharing partner. The risk of sharing data, then, can be defined as

$$\textit{data sharing risk} = \textit{data sensitivity} \cdot (1 - \textit{partner trust}) \quad . \tag{2}$$

For differential privacy, we further multiply this risk with the probability that the data sharing partner can guess the underlying data. If the probability is one, then the partner can guess the data and we have the standard risk formulation (2). If the probability is lowered, via use of differential privacy, then the overall data sharing risk is lowered, as well.

The trust and sensitivity parameters must be estimated subjectively by the data owner. This estimation process is out of scope of this paper, but it would occur prior to setting the noise level for differential privacy. One approach for producing these estimates is to break partner trust and data sensitivity into underlying factors that can be evaluated more objectively and then combining these factors back into overall scores. For example, Mayer et al. [19] broke trust in a partner organization into three factors: their ability to keep a secret, the benevolence of their organization toward ours, and their general integrity. Both the *data sensitivity* and *partner trust* parameters will thus receive a value between 0 and 1.

On the other hand, the probability of guessing is something that depends on the query type, as well as the DP mechanism in use. In particular, it can be tuned by the parameter of differential privacy. Formally, the attacker's goal is to guess the categorical attribute of a certain record in the database, e.g., learn the disease state of Alice. Differential privacy protects against strong attackers who already know the disease states of all other residents in the dataset, and, if we aim to use DP as our privacy protection mechanism, it makes sense to consider such attackers. Before the attacker has observed the output, he has already formed some opinion on the disease state of Alice, and has assigned some probability to the "correct" state of Alice. Let us call this the prior probability. Observing the output may change the attacker's opinion about the inputs and increase this probability. Let us call this the posterior probability. We can state our privacy measure as one of the following:

1. How large is the posterior probability?
2. How much larger is the posterior probability compared to the prior probability?

These privacy metrics in the context of differential privacy have been considered in [17]. Let us give formal definitions these two quantities. Let X be a random variable representing possible inputs from which the attacker may choose (i.e. possible choices for the underlying database), and let $\mathsf{supp}(X)$ be the support of X. Let $\alpha : \mathsf{supp}(X) \to [0,1]$ be the adversary's prior belief on X. Let M be a differentially private mechanism, and let $\gamma \in M(\mathsf{supp}(X))$. The posterior belief $\beta_\gamma : \mathsf{supp}(X) \to [0,1]$ on X after observing γ is defined as follows.

Definition 2.2 (Posterior belief on $X = x$ [17]) *For each possible choice $x \in \mathsf{supp}(X)$, the adversary's posterior belief on x is defined as*

$$\beta_\gamma(x) := \Pr[X = x \mid \gamma] = \frac{\Pr[M(x) = \gamma]}{\sum_{x' \in \mathsf{supp}(X)} \Pr[M(x') = \gamma]} \ .$$

In general, $\mathsf{supp}(X)$ may be the set of all possible states of the entire database, and x be any possible instance of the database. We are assuming a strong attacker who already knows all database records except the one that he is trying to guess. In the running SIRD example, this makes $\mathsf{supp}(X)$ a set of four possible choices of the disease state for the victim individual. The quantity $\beta_\gamma(x)$ can be viewed as the probability of disclosing x after seeing γ.

The adversary issues a query against the database and gets a noisy answer. After seeing the query response, the adversary computes the posterior belief for each possible choice $x \in X$. Finally, the adversary selects one x with the highest posterior belief as the "best guess".

Definition 2.3 (Posterior guessing probability) *The adversary's posterior probability of guessing the value of X is defined as*

$$\max_{\gamma \in M(\mathsf{supp}(X)), x \in \mathsf{supp}(X)} \beta_\gamma(x) \ .$$

Definition 2.3 quantifies the probability of guessing directly, which seems a quite intuitive definition. The problem of this choice is that the same number can denote different severities of leakage, e.g., a posterior probability $\beta_\gamma(x) = 0.90$ is clearly bad if the prior probability has been $\alpha(x) = 0.01$, but it is fine for $\alpha(x) = 0.89$, where observing the output has almost not made any difference. Also, if $\alpha(x)$ is already large, then increasing noise magnitude will never make the guessing probability smaller than $\alpha(x)$, which may result in a high reported risk even if no data is released at all. In our running SIRD example, if we set *data sensitivity* $= 1$ and *partner trust* $= 0$, the risk will never descend below 0.25 regardless of the added noise. It is more relevant to estimate how much the posterior belief $\beta_\gamma(x)$ has been *increased* compared to the prior belief $\alpha(x)$.

Definition 2.4 (Guessing advantage) *The adversary's advantage in guessing the value of X is defined as*

$$\max_{\gamma \in M(\mathsf{supp}(X)), x \in \mathsf{supp}(X)} (\beta_\gamma(x) - \alpha(x)) \ .$$

Definition 2.4 fixes the lower bound of resulting privacy measure to 0, but the upper bound will not exceed $1 - \min_{x \in \mathsf{supp}(X)} \alpha(x)$, and can be hard to perceive, as it will be different for different priors α. We can normalize the latter value and get a value from the segment $[0,1]$ scaling the guessing advantage by $1 - \min_{x \in \mathsf{supp}(X)} \alpha(x)$. In this case, 0 will mean "no additional information gain", and 1 "full leakage".

For a given ε, we can estimate an upper bound on the posterior belief $\beta(x)$ as described in [17, 18]. The following discussion is specific to our running example, but it can be easily adapted for other database schemas and queries. In our example, we want to hide the disease state of a particular user, i.e., which of the four aggregated counts his record contributes to. Let us define the underlying distance as $d(x', x) = k$ iff the disease states of some k users are different for the databases x and x'. Such distance is reasonable for any categorical data.

First, let us consider one count query output, which allows us to directly use the results of [17]. As shown in Sec.5.1 of [17], for all $x \in \mathsf{supp}(X)$, we have

$$\beta_\gamma(x) = \frac{1}{1 + \frac{\sum_{x' \in \mathsf{supp}(X) \setminus \{x\}} \Pr[Z = \gamma - f(x')]}{\Pr[Z = \gamma - f(x)])}} \tag{3}$$

where f is the query (without adding noise), γ is the noisy output, and Z is the random variable representing added noise.

The probability distribution of noise depends on the used DP mechanism. As shown in [17], for Laplace mechanism we have

$$\frac{\Pr[Z = \gamma - f(x')]}{\Pr[Z = \gamma - f(x)]} \geq e^{-\frac{\varepsilon \Delta v}{\Delta f}}$$

where

- Δf is the query sensitivity, i.e., how much (at most) the query output would be different for another input x' such that $d(x, x') = 1$. We have defined $d(x, x') = 1$ iff the tables x and x' differ in some users disease state. Modifying a users disease state may change the output of a single histogram bar at most by 1, so $\Delta f = 1$.
- Δv is the maximum difference between $f(x)$ and $f(x')$ for all pairs (x, x') of inputs that the attacker considers possible as the true input [17]. As we assume a strong attacker that already knows all database records except the one that he is trying to guess, the possible inputs x and x' may differ only in one record. With our definition of distance for categorical data, we always have $d(x, x') = 1$ for such x and x', so $\Delta v = \Delta f = 1$.

In particular, for uniformly distributed X, for all $x \in \mathsf{supp}(X)$, $\gamma \in M(\mathsf{supp}(X))$, we get an upper bound on posterior belief

$$\beta_\gamma(x) \leq q := \frac{1}{1 + (n-1) \cdot e^{-\frac{\varepsilon \Delta v}{\Delta f}}} \ ,$$

where $n = |\mathsf{supp}(X)|$ is the number of choices that the attacker has (in our example, $n = 4$ for the states S, I, R, D).

We can generalize the result to several observed noisy outputs $\gamma_1 \dots \gamma_m$. In (3), instead of probabilities $\Pr[Z = \gamma - f(x')]$, we get

$$\Pr[Z_1 = \gamma_1 - f_1(x') \wedge \dots \wedge Z_m = \gamma_m - f_m(x')] \ .$$

Since noise is sampled for each histogram independently, this equals

$$\Pr[Z_1 = \gamma_1 - f_1(x')] \cdot \ldots \cdot \Pr[Z_m = \gamma_m - f_m(x')] \ .$$

As in the single output case, assuming that the same Laplace mechanism is applied to each output, for all $1 \le i \le m$ we have

$$\frac{\Pr[Z_i = \gamma_i - f_i(x')]}{\Pr[Z_i = \gamma_i - f_i(x)]} \ge e^{-\frac{\varepsilon\Delta v}{\Delta f}} \ ,$$

hence,

$$\frac{\Pr[Z_1 = \gamma_1 - f_1(x')] \cdot \ldots \cdot \Pr[Z_m = \gamma_m - f_m(x')]}{\Pr[Z_1 = \gamma_1 - f_1(x)] \cdot \ldots \cdot \Pr[Z_m = \gamma_m - f_m(x)]} \le e^{-\frac{m\varepsilon\Delta v}{\Delta f}} \ .$$

Overall, for n possible input choices and m outputs, we get an upper bound on posterior probability

$$q = \frac{1}{1 + (n-1) \cdot e^{-\frac{m\varepsilon\Delta v}{\Delta f}}} \ . \tag{4}$$

From [15], we see that the same upper bound can be obtained not only for Laplace mechanism, but also any other mechanism that ensures ε-differential privacy.

For our running example, we can actually get a better bound. Since changing one persons disease status changes at most 2 histogram bars (the person is removed from one and added to some other), we have $f_i(x') = f_i(x)$ and $f_j(x') = f_j(x)$ for some $1 \le i \ne j \le k$. This gives us $e^{-\frac{2\varepsilon\Delta v}{\Delta f}}$ instead of $e^{-\frac{m\varepsilon\Delta v}{\Delta f}}$.

2.2 Estimating the noise

2.2.1 Absolute error

In addition to risk, we need to take into account the amount of noise. A value sampled from the Laplace distribution is unfortunately unbounded, so the added noise can be arbitrarily large. However, there exists a finite span within which the added noise stays with some confidence p. While $p = 100\%$ would keep the error magnitude unbounded, we can fix in advance some reasonable probability, e.g., $p = 90\%$ or $p = 99\%$, and report the quantity A below which the added noise stays with probability p. Such A has to satisfy the equality $\int_{-A}^{A} g(x)dx = p$, where $g(x) := \mathrm{Lap}(1/\varepsilon)(x) = \frac{\varepsilon}{2} \cdot e^{-\varepsilon|x|}$ is the probability density function of the Laplace distribution with scaling $1/\varepsilon$. This equation has a nice solution $A = \frac{-\ln(1-p)}{\varepsilon}$, which allows the computation of $\varepsilon = \frac{-\ln(1-p)}{A}$ as the DP parameter that gives an error of magnitude A.

In some cases, we may still need strictly $p = 100\%$. While Laplace distribution is unbounded, we can use *truncated Laplace distribution*, chopping off the distribution tails and fitting the range of noisy outputs into a finite span. While we cannot achieve pure ε-DP, we can instead achieve (ε, δ)-DP, which will also give us bounds on posterior guessing probability.

Definition 2.5 *(Approximate Differential Privacy [10]) Let X be the set of all possible databases to which a query may be applied. Let $\varepsilon, \delta \ge 0$. A mechanism M is* (ε, δ)-differentially private *if, for any two database instances $x_0, x_1 \in X$, and for any subset $Y \subseteq M(X)$ of outputs, we have* $\Pr[M(x_0) \in Y] \le e^{\varepsilon \cdot d(x_0,x_1)} \cdot \Pr[M(x_1) \in Y] + \delta$.

If $\delta = 0$, then the mechanism is just ε-differentially private as in Definition 2.1. Intuitively, pure ε-DP fails with a small error probability, given by δ. Extending this idea to guessing advantage, it is possible (with a small probability) that the noisy output is "bad", and leaks everything about the input. We show that, using (ε, δ)-DP mechanism based on truncated Laplace distribution from [11], we can choose ε in such a way that the observed noisy output keeps the attacker's posterior guessing advantage below a certain bound with probability $1 - \delta$.

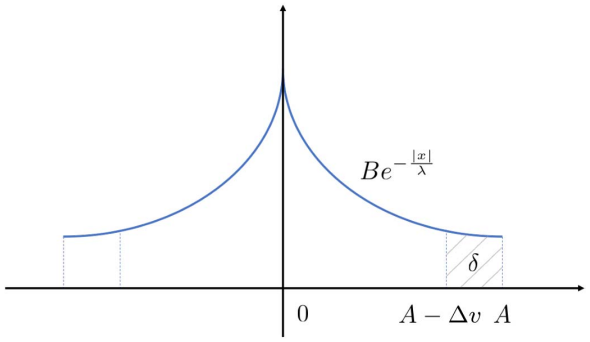

Figure 1: Noise probability density function of the truncated Laplacian mechanism [11]. Here A is the upper bound on noise magnitude, B the normalization factor, and $\Delta v = \Delta f \cdot d(x_0, x_1)$ the maximum possible change in the output.

Probability density function of truncated Laplace noise is depicted in Figure 1 (the image is taken from [11]). Let x be the true input, and let x' be another possible choice. Without loss of generality, let $f(x') \le f(x)$. The output $f(x')$ can be at most Δv apart from $f(x)$, so the distribution $\mathrm{Lap}(1/\varepsilon)(x')$ is the same as $\mathrm{Lap}(1/\varepsilon)(x)$ shifted at most by Δv. If the output gets into the range $[A - \Delta v, A]$, the attacker will clearly see that the input could not have been x'. However, for $Z \in [-A, A - \Delta v]$, we have $\frac{\Pr[Z=\gamma-f(x)]}{\Pr[Z=\gamma-f(x')]} \ge e^{-\frac{\varepsilon\Delta v}{\Delta f}}$, so for $\gamma \in [f(x) - A, f(x) + A - \Delta v]$ we can compute an upper bound on $\beta_\gamma(x)$ using (4).

Differently from (4), this upper bound is not directly applicable for any other mechanism that gives (ε, δ)-DP. However, it is applicable to any mechanism that satisfies

$$\forall y \in \mathbb{R}, x_0, x_1 \in X : \Pr\left[\frac{pdf_{x_0}(y)}{pdf_{x_1}(y)} \ge e^{\varepsilon \cdot d(x_0,x_1)}\right] \le \delta \ , \tag{5}$$

where pdf_x is the probability density function of $M(x)$. It has been shown in [5, Theorem 5 and condition (2)] that (5) is a sufficient condition for obtaining (ε, δ)-DP, and it is satisfied also by the Gaussian mechanism [5].

If the query result consists of m outputs, it is safe to assume that we are having the "bad" case if at least one output is "bad", which happens with probability $1 - (1 - \delta)^m$. Hence, when sampling the noise for the i-th output, it suffices to take

$$\delta' := 1 - \sqrt[m]{1 - \delta} \ , \tag{6}$$

which is the same for all outputs of the query result.

The construction of [11] allows us to compute the error upper bound A from ε and δ for truncated Laplace noise, assuming $\delta < \frac{1}{2}$. We now need to decide how to choose these two parameters, and there are several possible ways to do it.

So far, we aimed to achieve the desired upper bound on guessing advantage for *any* noisy output γ, which was a free variable in Def. 2.2. Using truncated Laplace mechanism, we fail to achieve this property for every possible γ, because certain output may exclude every input except one, and hence the posterior guessing probability will be 1. We will hence consider γ as a random variable, depending on the prior distribution α and the randomness of the added noise.

We may ask for a negligible probability of getting such γ that causes an upper bound for the posterior guessing probability to be violated. The probability of getting such γ is upper-bounded by δ. Since we consider the case where the attacker only gets a single

output, it is fine to take $\delta = 2^{-40}$. We then compute ε for the given guessing advantage as we did it for ordinary Laplace mechanism. A disadvantage of this approach is that fixing a negligible δ may result in high upper bounds on the error, even though the highest errors will actually come with negligible probability.

Truncated Laplace mechanism of [11] gives us the upper bound $A := \frac{\log(1+\frac{e^{\varepsilon}-1}{2\delta})}{\varepsilon}$ on the absolute error. In the visualization tool, we also need to compute ε from A and δ, i.e. solve

$$e^{\varepsilon} - 2\delta e^{A\varepsilon} \leq (1 - 2\delta)$$

for ε. This can be solved numerically, or a safe approximate solution can be computed.

Since the noisy output γ is itself a randomized value, instead of trying to protect the data for *any* γ, we could bound the *average* guessing advantage over the distribution of γ. This approach leads to smaller error magnitude.

2.2.2 Relative error

The noise estimate computed in either way can be arbitrarily large, and hence its goodness is difficult to interpret. The badness of A depends on the query, its result, and its further use by the recipient of the query result. For example, for "reasonable" use cases, the additive noise ± 5 would almost have no effect on the actual count $y = 1000$, but it would be destructive for $y = 10$. We assume that the data owner has a pretty good idea how the recipient is going to use the query result; hence he also knows the acceptability of each level of noise.

Instead of the absolute error, we may use the relative error. For a single count, the relative error is defined as $A/|y|$, where y is the true output. For a query result with m outputs, we can generalize this estimate to vectors as $\|A_1, \ldots, A_m\| / \|y_1, \ldots, y_m\|$, where $\|\cdot\|$ is the Euclidean norm. We note that we need to know the actual data to get the outputs $y_1, \ldots, y_m$, and if the data is not available, we have to use the absolute error. Even the relative error still does not have a fixed upper bound, as the noise may be several times larger than $|y|$. In fact, it is unbounded, as it approaches ∞ in the process $\varepsilon \to 0$; this process is a consequence of the process $q \to 0$. That is, an output that does not leak anything at all would in general be random noise itself.

We do not want to have the data owner fix an unbounded value, as there is no impression on how much error is bad enough. A possible solution here is to discretize the possible choices of q and set the smallest guessing probability to a fixed value, e.g., 0.01. This choice determines the largest possible relative error that the data owner may get so that any error above that value is considered extremely high and unreasonable. This solution results in asking the data owner to choose a relative error that has an upper bound of 100%. We may agree to bound the error by 100%, marking it as "very high", since a larger error would mean e.g., a query could return a response with a negative value for a count. In practice, a much lower upper bound of 50% might be a meaningful limit.

2.3 Summary

So far, we have described how to convert between the relative error and the desired probability of guessing (which in turn depends on the desired risk). The data owner's workflow would be to assess trust and sensitivity, then asses the maximum tolerable risk and noise to maintain utility, and then find a noise level that satisfies those constraints. In this workflow, there is a step where data owner has to convert the noise level to utility loss (or vice versa). This conversion is very much dependent on the underlying data, on the query, as well as on the usage of the results of the query by the analyst (as we discuss in Sec. 1). When all these quantities have been fixed, then the conversion could perhaps be automated. Right now, however, for the increased generality, we do not attempt to automate it, but assume that the data owner is by himself able to convert between the scales of added noise, and utility.

Let us summarize the discussion above in Table 1 as a collection of parameters that need to be defined in advance.

As an input, the system takes the maximum tolerable absolute noise A. There is a query being analyzed, with certain Δv and Δf computed once for the query. Assuming that δ has been fixed (e.g. $\delta = 2^{-40}$), the system computes internally the DP parameter ε. If relative noise A' is provided instead of absolute noise A, then the system should first evaluate the actual output y and compute $A = A' \cdot \|y\|$. The system then estimates the resulting risk value as

$$r = s \cdot (1 - t) \cdot \frac{1}{1 + (n-1) \cdot e^{-m\varepsilon \frac{\Delta v}{\Delta f}}} \quad ,$$

where m is the number of numeric outputs of the query, e.g., the number of histogram bars. As we have shown above, for some queries we can take a smaller value of k, resulting is smaller risk for the same ε. If the risk is assessed as tolerable, then ε is a suitable parameter, and noise sampled from $\mathrm{Lap}(\Delta f/\varepsilon)$ will be added to the query output. The quantities Δv and Δf depend on the query. For categorical data, we always have $\Delta v = \Delta f$, so $\Delta v/\Delta f = 1$, and we do not need to compute these quantities. We have not yet defined the parameter n anywhere, which is the total number of possible choices that the attacker may have. Ideally, this value should be read from the database schema, where categorical data is defined as an ENUM datatype of n possible values. If the query is defined over multiple data tables containing sensitive data, then we need to access privacy risk of each of these tables separately.

3 Differential privacy policy tool

We have developed a user interface called the differential privacy policy tool (DPP) that incorporates our new differential privacy mechanism. The visualization is designed to help a data owner decide how much noise to add to shared statistical data in order to protect the underlying data from being guessed, while still preserving the utility of sharing the data, all while remaining easy for a lay user to understand and use. This tool is part of a larger tool for assisting an officer responsible for setting up the information sharing policies of a large enterprise. The policy creation tool in turn is a part of a large experimental system for controlled information sharing in coalition operations. The tool enables the creation of unambiguous policies for sharing specific data with specific data sharing partners. These data sharing partners can then make queries, and those queries are evaluated against all current data sharing policies. Acceptable queries are then allowed to proceed. The DPP adds to the system the ability to make data sharing policies that include differential privacy.

To determine the appropriate level of noise for a differential privacy policy, the data owner must first establish the level of trust in the data sharing partner to not exploit the data and the sensitivity of the data underlying the counts. The data owner must also determine the maximum level of risk that can be tolerated and the maximum level of noise that can be tolerated while keeping the data meaningful and useful.

Here, trust, sensitivity, and risk are treated as qualitative assessments, running from very low to very high (mapped to values between 0 and 1, as in (2)), based on a variety of factors. It is assumed that the data owner has already determined the level of trust they have in their data sharing partner and the level of sensitivity of the underlying data. It is also assumed that the data owner has already assessed the maximum level of noise that can be tolerated while keeping the data useful and the maximum risk that can be tolerated. All that is left to consider is what percentage of noise to add to the shared data to meet the constraints on risk and utility.

In our running example of sharing SIRD counts, the underlying data, the SIRD state of individuals in each village is considered

Table 1: Differential privacy and data sharing risk parameters

Parameter	Lower Bound	Upper Bound	Meaning	Limitations
trust (t)	0%	100%	Subjective estimate of how much the party allowed to execute a DP query is trusted.	
data sensitivity (s)	0%	100%	Subjective estimate of how sensitive is the underlying data.	
maximum tolerable absolute noise (A)	0	∞	How far the noisy output is allowed to get from the true output.	There is no general reasonable upper bound.
maximum tolerable relative noise (A')	0%	100%	How far the noisy output is allowed to get from the true output. Compared to absolute error, 100% can be treated as a reasonable upper bound.	Need to know (an approximate) true query output to estimate.
probability of privacy failure (δ)	0%	50%	Probability that the added noise is sampled badly and does not protect sensitive data. Fixed to a statistically negligible value, e.g. 2^{-40}.	

highly sensitive because individuals could be persecuted. Therefore, using differential privacy to share the counts is recommended in order to limit the guessing of that underlying data. Second, the communications officer has assessed that the nations, overall, are low trust. Third, the maximum tolerable noise is about 5-10% because costly medical supplies need to be staged appropriately, and the maximum tolerable risk is low. Consequently, the communications officer needs to set an appropriate noise level, no higher than 10%, that achieves a data sharing risk of no more than low risk.

Figure 2 shows the DPP user interface. The data owner has selected low trust and high sensitive data on the left hand sliders and zero noise to add on the middle slider. The noise or relative error is translated into percent added noise on the user interface under the assumption that percent added noise is easier for a lay user to understand. The graph shows trust on the X axis, sensitivity on the Y axis, and risk on the Z axis. The surface in the graph shows the relationships among trust, sensitivity, and risk for the given percentage of noise added to the data.

The probability of guessing is not shown in the graph, but it is displayed as text below right of the graph. Changing the percentage of noise added to the counts changes the probability of guessing and the slope of the surface. In effect, adding more noise lowers the slope and results in lower probabilities of guessing and lower risk.

Given the selected levels of trust and data sensitivity, and no noise added, the risk of sharing the counts and having the underlying data guessed and exploited is medium. The yellow dot on the surface indicates this point in the space of trust, sensitivity, and resulting risk.

Figure 3 shows the effect of adding 10% noise to the shared data. The slope of the surface falls, the probability of guessing is reduced, and the risk from sharing these data drops to very low. This level of noise means that for a count of 100, the shared value would be some value between 90 and 110. If this level of noise is tolerable for maintaining the utility of sharing the data with the responding nations, then the communications officer has found an acceptable level of noise that keeps data sharing risk very low while maintaining data sharing utility. The officer could then accept this choice and complete the data sharing policy definition. In this way, users can modify the level of noise added and view the impacts on guessing probability and risk.

4 Focus group assessment

The design of the DPP tool was assessed by a focus group of human factors professionals. The objective was to evaluate both the concept and its meaningfulness to a naïve user as well as assess the tool designs usability.

Participants Eight human factors professionals participated in a series of three focus groups of 2-3 participants per group. The professionals were recruited from [Company affiliated with some of the authors], a human factors company, but they were all naïve to the project and user interface. The participants were each paid $40.

Materials The DPP tool was programmed in R and the R extensions Shiny and Plotty [21]. Three example situations were constructed where using differential privacy to share counts was plausible. The first example was the epidemic. The other examples varied the context (epidemic, genetic predictors, disaster evacuation) and the trust, sensitivity, noise tolerance, and risk tolerance.

Procedure Participants were introduced to the concept of differential privacy from the practical perspective of reducing the probability of the data sharing partner guessing the underlying data. They were then introduced to the DPP tool and led through the three example situations. Participants were encouraged to interact with the tool by trying different percentages of noise and observing the resulting levels of risk. They were also encouraged to explore different data sharing partners having different trust ratings. This idea was that perhaps an acceptable level of risk could be obtained if a better trusted partner were chosen.

Following the three examples, the participants were asked a short set of questions about the utility and understandability of the DPP tool. These questions opened up a discussion of design features, workflow, and usability. Finally, participants completed a system usability scale [7].

Results The participants offered a wide variety of feedback on the design of the tool and potentially better designs for supporting the data owner setting an appropriate level of noise. The mean SUS score was 71, and the scores ranged from 53 to 85. Designs having scores above 70 are considered acceptable [6].

The most critical feedback was to include a simpler 2D graph that would allow data owners to view the relationship between noise and risk directly, allowing the users to find a maximum tolerable level of risk and read off a percentage of added noise that would achieve that level of risk. This alternative graph better matches the users goal, and workflow, of determining a level of noise, while the current 3D graph is better conceived as a tool for further exploring the relationships of trust, sensitivity, noise, and risk. Participants also offered a variety of design suggestions. A major suggestion was

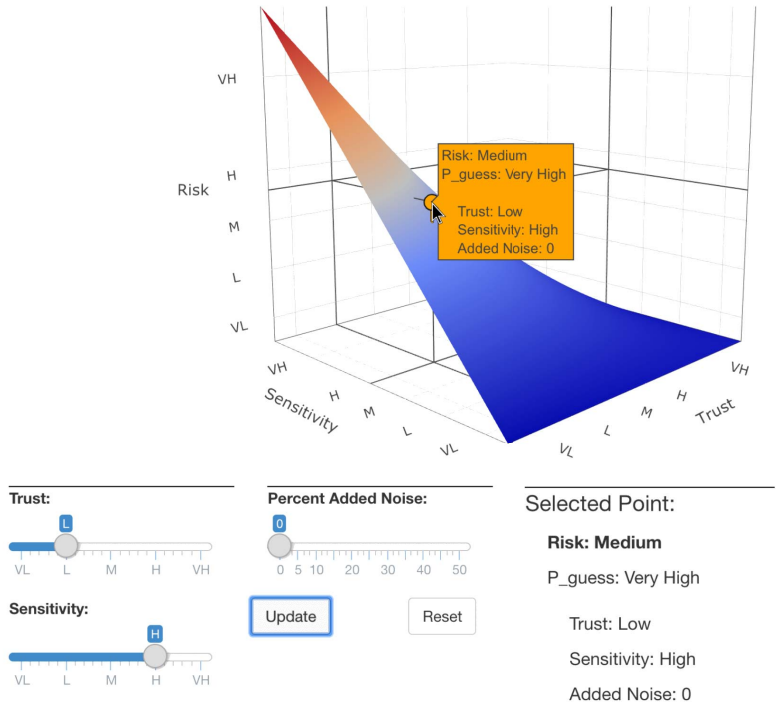

Figure 2: The interactive DPP tool with low trust, high data sensitivity, and no noise selected. The reticule indicates where the point hits each axis. The yellow hover-over call-out provides a read out of the trust, data sensitivity, and risk at the point where the mouse is located as it is moved around the surface.

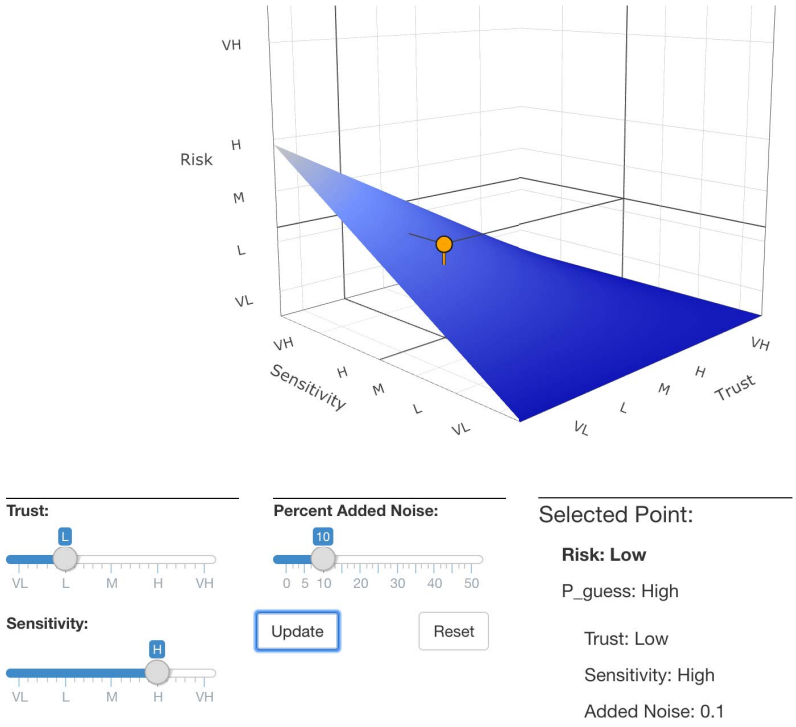

Figure 3: The interactive DPP tool with 10% noise added. The resulting risk for this combination of trust, sensitivity, and noise is very low.

to make the surface in the 3D graph translucent so that it would be easier to see the reticle and where it hits each axis, even if it were below the surface. Some participants were confused that the yellow color of the dot meant the point was medium risk, and they suggested that a more neutral color would be better. Another suggestion was to provide examples of what adding a percentage of noise means and a short, written summary of the chosen values and the resulting risk. These explanations could help naïve users better understand what their chosen percentage of noise meant, both in terms of what data values would be shared and what was the risk of the data sharing.

Discussion The feedback was used to develop a version 2.0 DPP tool. Most importantly, this tool incorporated the 2D noise x risk graph and better laid out the users workflow. The original 3D graph was maintained as a tool for further exploring trust, sensitivity, noise, and risk trade-offs. The design of the 3D graph was improved in accord with the feedback.

Version 2.0 of the DPP tool is shown in Figure 4. The upper left enumerates the sequence of steps for the data owner to follow: selecting trust and data sensitivity ratings, and choosing a maximum tolerable level of risk. The tool then calculates and graphs the trade-off between percent added noise and risk for the given data set. The graph also indicates the level of noise associated with the maximum tolerable risk. The data owner can then adjust noise to any desired level and see that point on the graph. The original 3D graph is available at the bottom of the display. The amount of noise chosen is explained in a tool tip, and a written summary of the chosen trust, data sensitivity, and noise was presented in the lower left. Once the data owner is satisfied with the level of noise added and the resulting data sharing risk, the data owner accepts the choices and proceeds to the next step in the data sharing policy creation process.

Four of the original eight participants were asked to review version 2.0. The participants were refamiliarized with the concept of the tool and walked through the first example, then they rated version 2.0 on the system usability scale. The average SUS score for version 2.0 was 89, a large improvement over version 1.0. The improvement was significant by a paired, one-tail t-test, $t(3) = 0.02$.

5 Comparison with related work

This paper is obviously not the first one to propose a manner in which good differential privacy parameters are chosen. The existing approaches [1, 12, 17] follow a common pattern. They cast both the privacy loss and utility loss in a common currency, and then minimize the total loss. This may indeed be the only tractable way, if the query is over data with a large number of owners. Our approach is different, in that we do not attempt to express the sharing risk and the amount of the noise in the same units. Instead, we present the Pareto-optimal choices to the data owner, who will be able to decide himself, which one he prefers the most.

We do express the sharing risk and the amount of noise in certain units; these or similar units have appeared before in the literature. The (average) posterior guessing probability, and the guessing advantage are the same as the *conditional min-entropy*, and the min-entropy leakage [23]. These quantities have been studied for differentially private mechanisms [4] showing relations very similar to Sec. 2.1. One considers the answering of the query and the subsequent addition of a noise as a channel from the set of inputs to the set of outputs, defined by giving for each possible input and each possible output the probability of getting this output under the condition that we start from the given input. Hence this approach naturally applies only to finite sets of possible inputs and outputs. To contrast this, in this paper, the set of possible outputs for our noised query is $\mathbb{R}$. Even though our set of possible inputs is discrete (though not technically finite), our approach is also applicable to continuous inputs, see Sec. 6.1 for discussion.

In the studies of min-entropy leakage, *gain functions* have been introduced, showing how valuable an "actual" guess or result is for a given "ground truth". The gain functions are used for both inputs [23], where they characterize the success level of a guessing adversary, as well as outputs [4], where they characterize the usefulness of the outcome for the analyst. The latter use is obviously related to the noise level. Alvim et al. [4] give an upper bound for utility, if the noise is added to the actual answer to the query, and the gain function for outputs is either Kronecker delta or at least highly symmetric. In this paper, we are using the same kind of noise, but our gain functions are different (and over a continuous domain). In fact, the proposed visualization mechanism could handle a wide variety of gain functions.

The min-entropy leakage has been computed or upper-bounded for various systems [8, 13, 22] and their classes. Rather less studied are the trade-offs between the bounds on min-entropy leakage of a mechanism, and the utility provided by this mechanism, as well as the optimization of one of those quantities under the bounds for the other one. Indeed, for optimization, there needs to be a (family of) parameters, the values of which affect the performance of the mechanism. In this paper, these are the parameters of the truncated Laplace mechanism. Ah-Fat and Huth [2, 3] have studied the optimization of min-entropy leakage under accuracy constraints. Their methods compute the "best" noise distribution for the given prior and the query using heavyweight optimization methods.

6 Future developments

6.1 From discrete to continuous data

While [17] allows easy estimation of q for categorical data and for a uniform prior, we can use the extended results of [15] to do the same for continuous data and different priors. In the case of continuous data, it does not make sense to let the attacker guess a value precisely, and guessing close enough can be bad as well. Instead, we need to define this sufficiently bad guessing radius as an additional parameter. While the radius is in general an unbounded quantity whose goodness is difficult to justify, we could normalize it, dividing by the maximum possible difference in two values. Such a relative radius would be a value ranging from 0 to 1, where 0 can be interpreted as a precise guess, and 1 as guessing nothing. This would require more inputs from the user, and we already have a lot of tuning parameters even for the simpler case.

Developing a user interface for privacy of continuous data could be viewed as a future work. This could be a two-step process, where the data owner first describes the sensitivity of the data. Such description basically introduces the gain functions also for inputs. The mechanism of the description could follow the structure of the database, building up a metric on it, perhaps using the notions of [16]. The second step would be similar to the procedure in this paper.

6.2 From one to several queries

So far, we have discussed how to quantify data leakage for a single query. In practice, the same analyst could make several queries to the same data. Similarly to generalizing a single output to a histogram, we could generalize the previous results even more, to multiple queries. For a fixed data disclosure risk, the magnitude of ε would be inversely proportional to the number of queries, similarly to the differential privacy sequential composition theorem [14]. To find a proper ε, we in general would need to know the number of queries in advance. If the data itself changes with time, then at some point it may become sufficiently independent from the old data, which may allow the use of less noise to achieve the same privacy guarantees. At the same time, if the data changes, then we need to clearly define the attacker goal for several database snapshots, e.g., is the attacker targeting a particular snapshot, or should we consider an attack successful if the private data has been guessed for at least one of the data snapshots. Details of generalization to several queries remains out of scope of this work.

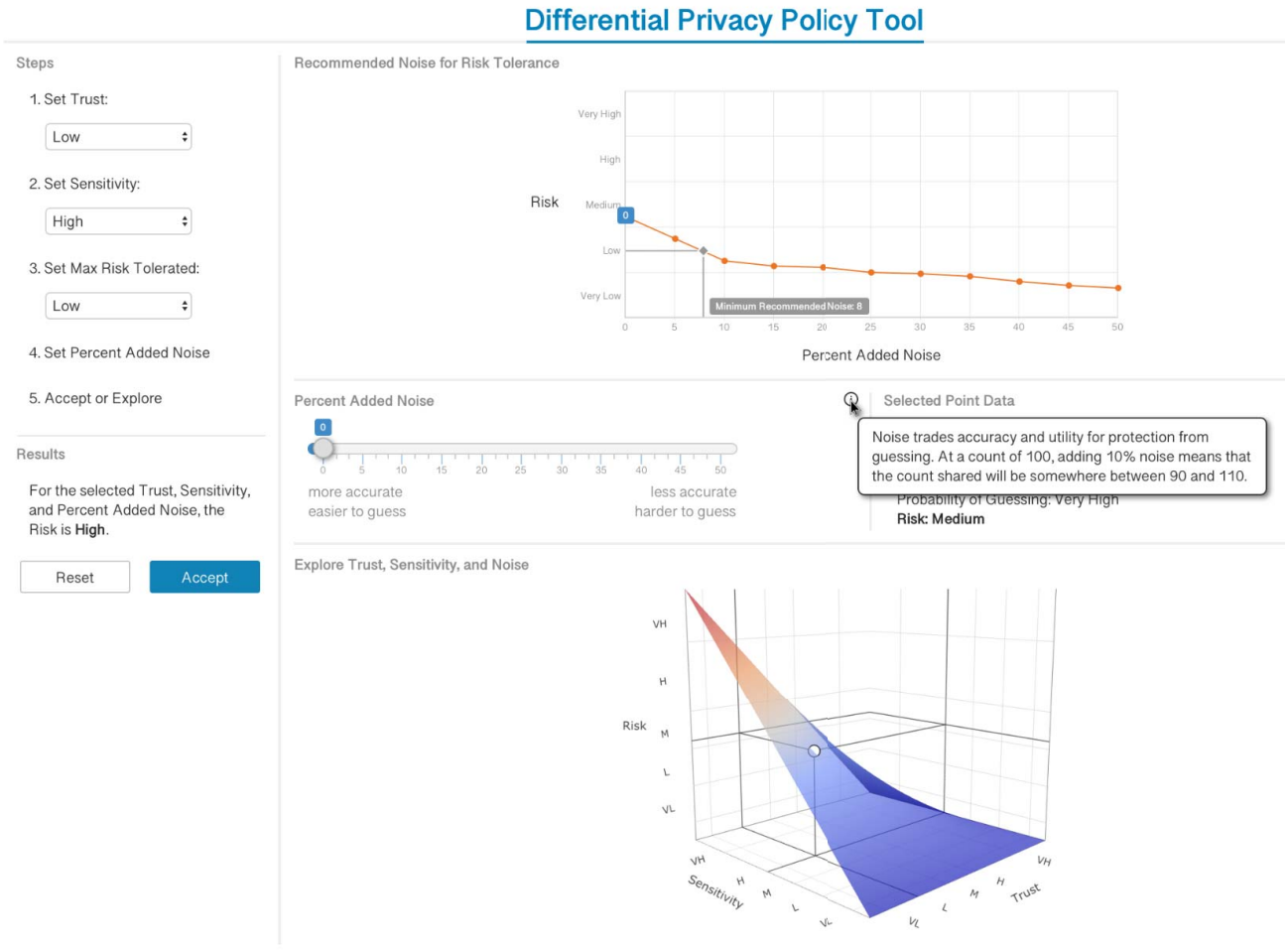

Figure 4: The revised DPP tool.

7 Conclusions

Differential privacy is a powerful technology for improving privacy while still sharing data. It may even promote additional data sharing because data owners can be more confident that the data they share will not be guessed and exploited.

The difficulty has been that differential privacy mechanisms require sophisticated, knowledgeable users to understand and set parameters. The goal of this research was to determine ways to minimize this burden so that even lay users would be able to configure and apply differential privacy technology in their data sharing practices.

This minimization required innovative thinking about the differential privacy mechanism and careful user interface design to employ that mechanism and procedure. The result is a usable user interface that requires little training, but that brings the power of differential privacy to practical use.

While the data owner must still make several subjective assessments about trust, data sensitivity, tolerable risk, and tolerable noise, the work represents an important step toward making provable data privacy techniques usable by the public.

References

[1] J. M. Abowd and I. M. Schmutte. An economic analysis of privacy protection and statistical accuracy as social choices. *American Economic Review*, 109(1):171–202, 2019.

[2] P. Ah-Fat and M. Huth. Optimal accuracy-privacy trade-off for secure computations. *IEEE Trans. Inf. Theory*, 65(5):3165–3182, 2019. doi: 10.1109/TIT.2018.2886458

[3] P. Ah-Fat and M. Huth. Protecting private inputs: Bounded distortion guarantees with randomised approximations. *PoPETs*, 2020(3):284–303, 2020.

[4] M. S. Alvim, M. E. Andrés, K. Chatzikokolakis, P. Degano, and C. Palamidessi. On the information leakage of differentially-private mechanisms. *J. Comput. Secur.*, 23(4):427–469, 2015. doi: 10.3233/JCS-150528

[5] B. Balle and Y. Wang. Improving the gaussian mechanism for differential privacy: Analytical calibration and optimal denoising. In J. G. Dy and A. Krause, eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, vol. 80 of *Proceedings of Machine Learning Research*, pp. 403–412. PMLR, 2018.

[6] A. Bangor, P. T. Kortum, and J. T. Miller. An empirical evaluation of the system usability scale. *International Journal of HumanComputer Interaction*, 24(6):574–594, 2008. doi: 10.1080/10447310802205776

[7] J. Brooke. SUS: A quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7, 1996.

[8] T. Chothia, Y. Kawamoto, and C. Novakovic. A tool for estimating information leakage. In N. Sharygina and H. Veith, eds., *Computer Aided Verification - 25th International Conference, CAV 2013, Saint*

Petersburg, Russia, July 13-19, 2013. Proceedings, vol. 8044 of *Lecture Notes in Computer Science*, pp. 690–695. Springer, 2013. doi: 10.1007/978-3-642-39799-8_47

[9] C. Dwork. Differential privacy. In M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, eds., *Automata, Languages and Programming*, pp. 1–12. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[10] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In S. Halevi and T. Rabin, eds., *Theory of Cryptography*, pp. 265–284. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.

[11] Q. Geng, W. Ding, R. Guo, and S. Kumar. Truncated laplacian mechanism for approximate differential privacy. *CoRR*, abs/1810.00877, 2018.

[12] J. Hsu, M. Gaboardi, A. Haeberlen, S. Khanna, A. Narayan, B. C. Pierce, and A. Roth. Differential privacy: An economic method for choosing epsilon. In *2014 IEEE 27th Computer Security Foundations Symposium*, pp. 398–410. IEEE, 2014.

[13] I. Issa, A. B. Wagner, and S. Kamath. An operational approach to information leakage. *IEEE Trans. Inf. Theory*, 66(3):1625–1657, 2020. doi: 10.1109/TIT.2019.2962804

[14] P. Kairouz, S. Oh, and P. Viswanath. The composition theorem for differential privacy. *ArXiv*, abs/1311.0776, 2013.

[15] P. Laud and A. Pankova. Interpreting epsilon of differential privacy in terms of advantage in guessing or approximating sensitive attributes. *ArXiv*, abs/1911.12777, 2020.

[16] P. Laud, A. Pankova, and M. Pettai. A framework of metrics for differential privacy from local sensitivity. *PoPETs*, 2020(2):175–208, 2020.

[17] J. Lee and C. Clifton. How much is enough? choosing ε for differential privacy. In X. Lai, J. Zhou, and H. Li, eds., *Information Security*, pp. 325–340. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[18] J. Lee and C. Clifton. Differential identifiability. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD 12, p. 10411049. Association for Computing Machinery, New York, NY, USA, 2012. doi: 10.1145/2339530.2339695

[19] R. C. Mayer, J. H. Davis, and F. D. Schoorman. An integrative model of organizational trust. *The Academy of Management Review*, 20(3):709–734, 1995.

[20] A. Narayanan, J. Huey, and E. Felten. A precautionary approach to big data privacy. *Data Protection on the Move: Current Developments in ICT and Privacy/Data Protection*, pp. 357–385, 2016. doi: 10.1007/978-94-017-7376-8_13

[21] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2019.

[22] D. M. Smith and G. Smith. Tight bounds on information leakage from repeated independent runs. In *IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 318–327. IEEE, 2017.

[23] G. Smith. Recent developments in quantitative information flow (invited tutorial). In *30th Annual ACM/IEEE Symposium on Logic in Computer Science*, pp. 23–31. IEEE, 2015.