Applications and Implementation of Differential Privacy on Statistical Databases

Jonathan Sumner Evans*, Victoria Girkins† and Sam Sartor‡
Department of Computer Science, Colorado School of Mines
Golden, Colorado
Email: *jonathanevans@mines.edu, *vgirkins@mines.edu, †rdmerillat@mines.edu,

***Abstract*—Statistical databases have long been know to be vulnerable to inference attacks. We describe the nature of inference attacks and one method of mitigating these attacks called Differential Privacy. We describe our implementation of a simple Differential Privacy statistical database and then performed experiments using the database to analyze how well our implementation achieves the objectives of Differential Privacy databases.**

I. Introduction

Statistical databases are designed for statistical analysis of the datasets contained within the database. One of the desired properties of statistical databases is that nothing about an individual should be learnable from the database that cannot be learned without access to the database [3]. In other words, gathering information about the records underlying the statistics using allowed queries on the database should be impossible.

Without any protections, well-crafted queries to a statistical database can implicitly reveal private information. This attack is know as an *inference attack*. To prevent these attacks, the key case to consider is that of two databases which differ by only a single row. Consider, for example, a database which associates people with their incomes. The query

```
SELECT SUM(income) FROM table
```

seems harmless at first. But if a new record were added and we ran the query again, the difference between the two results represents the exact income of the new entry. If the attacker knows the identity of the person who was added, the privacy leak becomes even more severe.

Of course, an attacker may not have the luxury of always knowing when a new record will be added to the database, and of timing his queries accordingly. He may, however, achieve the same results by narrowing down his query with conditionals.

```
SELECT SUM(income)
  FROM table
 WHERE zip != 80127
```

may be sufficient in a small enough database, or more conditionals may need to be added to ensure the queries jointly isolate a single row. However, since the danger occurs when information about a single entry is returned, regardless of how the attacker managed it, we will assume for simplicity's sake that the database is queried before and after the addition of a single row.

A. Previous Attempts at Anonymizing Statistical Data

Many attempts have been made to privatize data in the past and many are still in use today. However everything from simply removing columns containing personally identifiable information to advanced techniques such as as k-anonymity and l-diversity have been shown to be vulnerable to attack [2]. K-anonymity, for example, does not include any randomization, attackers can still make inferences about data sets [1].

B. Differential Privacy

In 2006, Cynthia Dwork proposed a new technique to reduce the risk of a successful inference attack called *Differential Privacy* [5]. The goal of Differential Privacy is to maximize the statistical accuracy of queries on a statistical database while also maximizing the privacy of the individual records. It does this by introducing noise into the dataset. This randomness reduces the likelihood of obtaining meaningful information about the individual records in the database.

The real power of Differential Privacy is when it is used in conjunction with a privacy bound, ε as described in Section I-B1. The methods for generating random noise are discussed in Section I-B2. Details about how the one of the mechanisms is used to ensure differential privacy are explained in Section I-B3.

1) Epsilon-Differentially Private: One useful way of discussing Differential Privacy is to talk about what it means for a database to be *epsilon-differentially private*. A database falls under this category if it is private as bounded by a chosen value epsilon. As opposed to the mathematical definition, here we give a more intuitive definition as stated by Atockar [2]:

> This [epsilon-differentially private] can be translated as meaning that the risk to one's privacy should not substantially (as bounded by epsilon) increase as a result of participating in a statistical database.

For the sake of rigorousness, the formal mathematical definition of epsilon-differential privacy described by Dwork [3] follows.

Definition 1. *A randomized function $\mathcal{K}$ gives* **ε-*differential privacy*** *if for all data sets D and D' differing on at most one row, and all $S \subseteq Range(\mathcal{K})$,*

$$\Pr[\mathcal{K}(D) \in S] \leq \exp(\varepsilon) \times \Pr[\mathcal{K}(D') \in S] \quad (1)$$

For now, all that concerns us about this equation is that D and D' are the above-mentioned databases differing by a single row, and $\mathcal{K}$ is our mechanism for adding noise drawn from our distribution as described in Section I-B2.

2) Differential Privacy Mechanisms: There are two primary Differential Privacy Mechanisms: the Exponential Mechanism and the Laplace Mechanism. The Exponential Mechanism should be used on categorical, non-numeric data. For example queries of the form "what is the most common eye color in this room" should be protected using the Exponential Mechanism [2]. The Laplace Mechanism should be used with non-categorical, real-value data [4]. Because of the nature of our data, our primary focus for this project is the Laplace Mechanism.

Before we rigorously examine the mathematics, let us review our inputs and our goals:

Inputs A statistical database with sensitive information, an attacker poised with a malicious query, and a desired epsilon, or privacy parameter.

Goals To add enough Laplace-distributed noise to the results of database queries that our database is mathematically considered differentially private.

Mathematically, the Laplace Distribution is defined as follows:

Definition 2. *The Laplace Distribution, $Lap(\mu, b)$, also known as a symmetric exponential distribution, is the probability distribution with p.d.f.:*

$$P(x|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|x-\mu|}{b}\right) \quad (2)$$

where μ is the location of the center of the distribution and b is the scale parameter.

For us, μ is 0 so we can use a variant of the Laplace Distribution called the zero-centered Laplace Distribution which has only one parameter, b, the scale.

3) Using the Laplace Differential Privacy Mechanism: In choosing the value for b, it makes sense that we should consider ε, which quantifies the level of privacy we desire. Another factor comes into play, however: the "sensitivity" of the database. This is defined as the maximum difference between the results of any query run on two databases which differ by only one row, as described above [2]. We shall assume

```
SELECT SUM(income) FROM table
```

is the only query that this database will accept, so the "sensitivity" of our database is clearly the highest income in the table, since this value will be the difference between two queries run on the database before and after the record associated with this income is added.

So, we must add noise while considering both ε and the database's above-defined sensitivity. In fact, in a paper by Dwork [**Dwork:2011**], it was proven that choosing the "scale" parameter (b) of our Laplace distribution according to this equation

$$b = \Delta f / \varepsilon \quad (3)$$

where Δf is sensitivity, our database will mathematically satisfy epsilon-differential privacy. That is, we should add noise to the data drawn from a 0-centered Laplace distribution with scale parameter $\Delta f / \varepsilon$.

The one further consideration we must make is the possibility of multiple queries to the database. The Laplace distribution is symmetrical, so if unlimited queries were allowed, the attacker could simply run his query many times and take the average to extract sensitive data. So we must limit the number of queries allowed, and fortunately the calculation is linear. Assume we draw all noise from the same distribution (it would be possible to vary it but this unnecessarily complicates things), and that this distribution is Laplace, 0-centered, with a privacy parameter ε_i (that is, with scale parameter $\Delta f / \varepsilon_i$). If we limit learners to k queries, our overall privacy budget is $k\varepsilon_i$. This privacy budget is our actual

epsilon; as stated by Atockar, "it reflects the maximum privacy release allowable for the total query session" [2].

So to sum it up practically: if we wish to make our database epsilon-differentially private as bounded by some value epsilon, our database has sensitivity Δf, and we wish for learners to be allowed k queries before being locked out, we should add a random variable to each query, drawn from the 0-centered Laplace distribution with scale parameter $k \times \Delta f / \varepsilon$.

Of course, the lower of a privacy budget we allow, the noisier the data returned by the queries will become [2]. If our queries become useless due to the noise in the data, we may choose to raise the privacy budget—that is, to increase the value of epsilon—or to further limit the number of queries each user is allowed—that is, to lower the value of k. As with many issues in information security, this is a balancing act between security and usability, and we should consider the nature of the database, its contents, and its purpose when choosing where to draw this line.

II. Approach

III. Experiment

A. Dataset

B. Crafted Queries

C. Database Implementation

IV. Results

V. Limitations and Future Work

VI. Conclusion

References

[1] Charu Aggarwal. "On k-Anonymity and the Curse of Dimensionality." In: *VLDB 2005 - Proceedings of 31st International Conference on Very Large Data Bases*. Vol. 2. Jan. 2005, pp. 901–909.

[2] Atockar. "Differential Privacy: The Basics". In: (2014). URL: https://research.neustar.biz/2014/09/08/differential-privacy-the-basics/.

[3] Cynthia Dwork. "Differential Privacy". In: *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*. Vol. 4052. Springer Verlag, July 2006. ISBN: 3-540-35907-9. URL: https://www.microsoft.com/en-us/research/publication/differential-privacy/.

[4] Quan Geng and Pramod Viswanath. "The optimal mechanism in differential privacy". In: *2014 IEEE International Symposium on Information Theory* (2014). DOI: 10.1109/isit.2014.6875258. URL: https://arxiv.org/pdf/1212.1186.pdf.

[5] Michael Hilton. *Differential Privacy: A Historical Survey*. URL: http://www.cs.uky.edu/~jzhang/CS689/PPDM-differential.pdf.