Enhancing Metric Privacy With a Shuffler

Andreas Athanasiou
INRIA and LIX, École Polytechnique
Palaiseau, France
andreas.athanasiou@inria.fr

Konstantinos Chatzikokolakis
National and Kapodistrian University of Athens
Athens, Greece
kostasc@di.uoa.gr

Catuscia Palamidessi
INRIA and LIX, École Polytechnique
Palaiseau, France
catuscia@lix.polytechnique.fr

Abstract

Differential Privacy (DP) is one of the most successful privacy-preserving frameworks. In the central model of DP a trusted server adds controlled noise as it acts as an interface between the data providers (users) and the data consumers (analysts). To overcome the strong trust assumption of having a trusted server, Local Differential Privacy (LDP) has been proposed, where the individual data are obfuscated directly at the end of the data provider. To improve LDP, in recent years researchers have proposed to combine it with a *shuffler* which is supposed to mix the data at the time of collection, enhancing the privacy of LDP without affecting utility. The shuffler is assumed to be trusted, but this is also an arguably strong assumption that cannot always be guaranteed.

Metric privacy (aka d-privacy) is a variant of DP that can be applied in domains provided with a notion of distance and it is particularly used in location privacy, where it takes the name of *geo-indistinguihability*. In contrast to DP, metric privacy allows calibrating the noise so that data points closer to the true one are more likely to be reported.

In this work we study how metric privacy can be improved by combining it with a shuffler. More specifically, we consider the combination of the shuffler with three mechanisms, Randomized Response, Geometric and an optimal protocol, in the context of the sum and average queries. In all cases, we formally derive the relations that express the privacy amplification due to the shuffler, in terms of metric privacy. Moreover, we formally study the privacy guarantees of each protocol if the shuffler is compromised. Finally we conduct experiments using synthetic data as well as real-world location data, showing that the proposed mechanisms achieve a better privacy-utility trade-off compared to the baseline of the standard geometric mechanism.

Keywords

Differential Privacy, Metric Privacy, Shuffle Model, Randomized Response, Geometric Mechanism

1 Introduction

Differential privacy (DP) [17] [16] is a formal notion of privacy, ensuring the protection of each individual in a statistical database, when it is queried by an analyst to get aggregated information. DP establishes a bound on the ratio of the probability to get the same answer from two adjacent databases, namely, two databases that differ for just one record. The bound is expressed in terms of a parameter ε, which represents the level of privacy. Most research focuses on two opposite approaches to implement DP. The first one is the central model of DP, where a *trusted* central party collects the data and injects noise, high enough to reach the desired level of privacy but also maintain the utility of the data. The central model requires to assume that the central party is trusted. This is the Achilles' heel of this approach.

At the other extreme, there is the local model [27] where the data providers (called users in this paper) apply noise on their own by using a so-called *Local Randomizer*. While this method enjoys a lack of obligation to trust a central party, it comes with its limitations, as certain learning tasks that can be performed in the central model cannot be performed in the local model [27]. Moreover, it is proven that in most cases, to achieve the same level (i.e. ε) of DP, the amount of noise needs to be higher, hurting utility [6, 8]. For this reason, often a high number of users is necessary to decrease the (relative) error. Therefore this model has been used mainly by technology giants such as Google [19], Microsoft [13] and Apple [12].

With the purpose of improving the trade-off between privacy and utility in the local model, A. Bittau et al. proposed to combine it with a *shuffler* [7]. In *the shuffle model*, users still run a Local Randomizer to locally add noise, but instead of sending the result directly to the central party, they send it to a middle layer, the shuffler. After shuffling (i.e. mixing) the inputs received from all the users, the shuffler sends them to the central party which can then query them. The idea is that shuffling provides an additional layer of obfuscation by eliminating the link between the data and their original owners. Hence, to achieve the intended level of privacy, users can add less noise to their data. Since the shuffler does not alter the utility of the data (at least for the queries that are commutative on the messages), the net result is an improvement of the privacy-utility trade-off. In conclusion, the shuffle model is an appealing compromise between the poorer utility of the local model and the stronger trust assumption of the central model.

In the shuffle model it is customary to assume that the shuffler is trusted. Indeed, this is a weaker trust assumption compared to the central model since the shuffler can be implemented in a distributed way. Furthermore, even if the shuffler is compromised, the data available to it still contain some noise, unlike the central model.

The distributed shuffling can be implemented via MPC [11], trusted hardware [7] or using a MixNet. A MixNet is a sequence of servers where each one receives the data, shuffles it and sends it to the next server. Zero-knowledge proofs need to be shared at each step to ensure that the server has not tampered with the data. The shuffling will be executed correctly as long as at least one server is honest. Since we cannot always assume that the first server is

Proceedings on Privacy Enhancing Technologies 2025(2), 650–679

https://doi.org/10.56553/popets-2025-0081

honest, an "onion encryption" protocol needs to be adopted (similar to Tor [14]). This type of encryption works by having the first server decrypt the first, "outermost layer" of encryption without having the key to decrypt the next layer, which is held only by the next server. The server has to now shuffle the encrypted messages. In other words, this ensures that no server in the MixNet can actually see the raw data.

Nonetheless, in real-world applications, the assurance that the shuffler is trusted is not always easy to achieve, which raises the question what are the consequences in a scenario where a shuffle model mechanism is used and the shuffler becomes compromised. If this happens, the mechanism essentially reduces to providing only local model privacy guarantees, and in fact with reduced privacy level (since the noise added by each user is typically reduced). Interestingly, however, not all shuffle model mechanisms behave in the same way wrt their local model privacy guarantees; some achieve their *privacy amplification* in a way that is somewhat "compatible" with local privacy, while others achieve it in a way that severely violates local privacy. Hence, as we see later in this paper, we can have scenarios in which two mechanisms have identical shuffle model privacy, comparable utility, but drastically different local model privacy.

However, in many applications one wishes to protect the *precision* with which an adversary can learn the user's secret. For instance, consider a scenario where a single individual, traveling in a city, wishes to use a location-based service while still enjoying a level of privacy. The adversary should not be able to guess his *exact* location, but could be able to deduce the district he is in. In other words, nearby secrets (locations) should look identical to the adversary, but far-away locations are allowed to be distinguished. Metric privacy was introduced in [9] (where it was called d-privacy) as a variant of DP suitable for domains provided with a notion of distance. Like in central and local DP, metric privacy imposes a bound on the probability that the same result is obtained from two different elements. However, in contrast to DP, this bound does not depend only on the parameter ε, but also on the distance between the objects. This means that the noise can be calibrated depending on how large the range in which we want to achieve indistinguishability is. In contrast, LDP requires indistinguishability between any pair of elements in the domain. Therefore metric privacy is particularly useful in those applications in which hiding an element within a group of neighbors is a sufficient measure of privacy protection.

Metric privacy has been applied mostly in the local model, and particularly for location privacy, where it takes the name of *geo-indistinguishability* [1]. However, metric privacy can also be applied in the central model; indeed, metric privacy can be considered an extension of both local and central DP. In the central case, the elements (arguments of the mechanism) are datasets, and the notion of distance that corresponds to the DP property is the Hamming distance (two datasets are adjacent if their Hamming distance is 1). Metric privacy, however, can also use different distances on datasets. In this paper we employ the Manhattan and the Euclidean distance, in which the level of distinguishability between two datasets not only depends on the number of different values but also on the values themselves.

A natural question, explored in this paper, is whether metric privacy can be applied in the shuffle model to provide a privacy boost similar to that of standard DP. Interestingly, although for standard DP we can take the natural local model mechanism (k-Randomized Response) and achieve a boost simply by shuffling the users' values, the same is not true for metric privacy. The natural mechanism in this case is the *geometric mechanism*, but simply shuffling its outcome does not achieve a non-negligible boost. The reason is that the probability of reporting a value close to the original one is much larger if we are only interested in metric privacy, which means that outlier values will be recognizable after the shuffling, in a sense canceling its effect. Hence, to achieve a privacy boost we need to employ the geometric mechanism in a non-trivial way; in fact, later in this paper we present two distinct such adaptations of the geometric mechanism, *Geo-Shuffle* and *SGDL-Shuffle*, with distinct privacy properties. To the best of our knowledge, this is the first work that proposes and investigates the shuffle model for metric privacy.

1.1 Contributions

- We study *RR-Shuffle*, a shuffle model variant of the Randomized Response mechanism, and derive its metric privacy and utility guarantees (Section 3.2).
- We propose *Geo-Shuffle*, a novel adaptation of the geometric mechanism (the natural choice for metric privacy). We show that its privacy can be expressed in terms of the Symmetric Generalized Discrete Laplace (SGDL) distribution which, as far as we are aware, has not been previously studied in the context of privacy (Section 4).
- We propose *SGDL-Shuffle*, a different adaptation of the geometric mechanism, based on techniques from [21]. We again study its privacy guarantees and show that it provides optimal utility (Section 5).
- We experimentally evaluate all proposed mechanisms in two datasets: a synthetic one and a real-world dataset of location data. We show that all mechanisms of the shuffle model provide significantly better utility compared to those of the local model. RR-Shuffle is outperformed by Geo-Shuffle and SGDL-Shuffle; the latter provides optimal utility that matches that of the central model, while the former closely approaches it (Section 6).
- We study the privacy guarantees of all mechanisms in the case where the shuffler is compromised. This type of analysis is often missing from the shuffle model literature. We show that, surprisingly, although Geo-Shuffle and SGDL-Shuffle have identical shuffle model privacy and comparable utility, their local model privacy is drastically different (Section 7).

1.2 Background and Related Work

For standard DP, the shuffle model has been studied in a series of works. Albert Cheu et al. [11], provided a formal analytical model by providing a protocol for Boolean Sums and a protocol for Real Sums of bounded input, using the famous mechanism of Randomized Response, proposed by Stanley L. Warner in 1965 [31]. For privately estimating bounded real-value statistical queries, they proposed a mechanism in the shuffle model which is (ε, δ) - differentially private with an error that is only $O(\frac{1}{\varepsilon} log \frac{n}{\delta})$, when executed with n users. This is close to the error of the curator model which is

$O(1/\varepsilon)$. For binary sums, the error reduces to $O(\frac{1}{\varepsilon}\sqrt{log\frac{1}{\delta}})$ and each user sends only one bit. Erlingsson et al. [18] proved that adding a shuffler to any local differential privacy protocol amplifies the privacy parameters. They showed that if a protocol satisfies ε - differential privacy in the local model then adding a shuffler will make the protocol satisfy $O(\varepsilon\sqrt{log(1/\delta)/n}, \delta)$ - DP in the central model.

Another interesting study of the shuffle model for standard DP was conducted by B. Balle et al. [4]. They proposed a single-message protocol of the shuffle model where estimating the sum of real numbers in $[0, 1]$, each held by one of n users results in an error with standard deviation of $O(n^{1/6})$. Each user sends only one message with size $O(logn)$, which results in a smaller communication cost compared to [11]. Moreover, they proved a lower bound of the error of any single-message protocol of the shuffle model. They showed that the mean square error has to be in the order of $\Omega(n^{1/3})$. Another interesting contribution is their proof of the privacy amplification that occurs when a shuffler is added. Shuffling n copies of ε_0- differentially private Local Randomizers each with $\varepsilon_0 = O(log(n/log(1/\delta))$ yields an ε, δ - differentially private mechanism with $\varepsilon = O(min(\varepsilon_0, 1)e^{\varepsilon_0}\sqrt{log(1/\delta)/n})$. An optimal error for the problem of private summation has been proved by B. Balle et al [5] and B. Ghazi et al. [22]. They showed that there is an (ε, δ) differentially private protocol in the shuffle model that has an error equal to the Discrete Laplace mechanism of the central model with parameter $(1-\gamma)\varepsilon$ for every $\varepsilon \leq O(1)$ and every $\delta, \gamma \in (0, 1/2)$. They also extended this theorem to the problem of privately computing histograms. In the recent work of Albert Cheu [10] an overview of all shuffle protocols is presented, for Binary Sums, Histograms, Uniformity Testing and Pointer Chasing problems. Also, he proves that the shuffle model, under natural constraints, is weaker, in terms of error, compared to the central model.

Note that the corruption of the shuffler has been studied in the literature. As an alternative approach, the framework of *Differentially Oblivious Shuffling* [24] was proposed as a way to reduce the trust on the shuffler by tolerating a small amount of information being leaked during shuffling (formally described by ε, δ, similarly to standard DP). In this paper, on the other hand, we study the case of the shuffler getting fully compromised, in which the shuffle's model privacy reduces to that of the local model.

The lack of studies on the application of the shuffle model in metric privacy is the motivation of this work.

1.3 Plan of the Paper

The next section recalls the preliminary notions of central and local DP, metric privacy and the shuffle model.

In Sections 3, 4 and 5 we discuss the proposed mechanisms which are based respectively on the Randomized Response mechanism, the Geometric mechanism and the Symmetric Generalized Discrete Laplace distribution.

Next, in Section 6 we evaluate the utility of the proposed shuffler-enhanced mechanisms, and compare them with the geometric mechanism in the local model.

In Section 7 we study how the proposed mechanisms compare, in terms of privacy, when the adversary also controls the shuffler.

Table 1: Table of Notations

Notation	Description
R	Local Randomizer
S	Shuffler
A	Analyst
d_X	Function measuring the distance between two datasets
d	The distance between two datasets (the result of d_X)
n	Number of users
k	Users have values in $\{0, \ldots, k\}$
x_i	Secret of user i
$\mathcal{U}$	Unary encoding (Algorithm 1)
λ	The expected number of random bits in RR-Shuffle
N_i	Noise of user i
c	Padding value
G	Geometric Mechanism
$\mathcal{G}$	Geometric distribution
NB	Negative Binomial distribution
ε_{geo}	ε of the local randomizer of Geo-Shuffle
Geo-Central	Geometric Mechanism in the Central Model
Geo-Local	Geometric Mechanism in the Local Model
ε_L	ε in the Local Model (shuffler compromised)
ε_S	ε in the Shuffle Model (shuffler not compromised)

Then, in Section 8 we justify our assumption that only a primitive shuffler can be used and discuss the communication cost of the proposed protocols. Finally, Section 9 concludes, and discusses future work.

2 Preliminaries and Definitions

2.1 Differential Privacy

We define a dataset as a vector $X = (x_1, \ldots, x_n)$ of n users, each with a numerical value, who wish to privately compute the sum of their values. In the *central model* of DP a curator has access to the full dataset X, and applies a probabilistic mechanism M which typically computes the outcome of the query and adds random noise to it. Two datasets X, X' are considered *adjacent*, if they differ by the value of exactly one element, or equivalently their Hamming distance is 1. Differential Privacy requires that such changes should not be observable in the query outcome.

Note that this corresponds to the so-called *bounded DP*. The standard definition of DP uses a different notion of adjacency, in which two datasets are adjacent if one can be obtained from the other by adding or removing one user. For our work, however, this definition is not convenient since the number of users in the shuffle model is observable, hence it is considered fixed.

Definition 1 (**Approximate Differential Privacy**). *A mechanism M is (ε, δ) - approximate differentially private iff for all pairs of adjacent datasets X, X' and all sets of results S:*

$$\mathbb{P}[M(X) \in S] \leq e^{\varepsilon} \cdot \mathbb{P}[M(X') \in S] + \delta$$

where the probabilities are in the randomness of M. If $\delta = 0$, then M is ε-differentially private.

A useful property of differential privacy is that any post processing operation does not affect privacy.

Lemma 2.1 (Post-Processing). *[16] If M is (ε, δ) - approximate differentially private then for every function A, $A \circ M$ is (ε, δ) - approximate differentially private.*

To overcome the strong trust assumption of the fully-trusted entity holding all users' data of the central model, the *local model* has been proposed. Each user, holding a value x from some domain $\mathcal{D}$, applies the mechanism M on his own data x, and publishes the mechanism's noisy result, thus avoiding the need for a central entity. The noise has to be large enough so that any particular value has substantial probability to be reported, leading to the following definition:

Definition 2 (**local model of Differential Privacy**). *[27] A mechanism M provides (ε, δ) - approximate Local Differential Privacy iff for all values $x, x' \in \mathcal{D}$ and all sets of results S we have:*

$$\mathbb{P}[M(x) \in S] \leq e^{\varepsilon} \cdot \mathbb{P}[M(x') \in S] + \delta$$

Although the local model achieves privacy without a central authority, the need to add noise directly to each value results in worse utility compared to the central model. As a consequence, the *shuffle model* [11] has recently emerged, with the goal of achieving the best of both worlds.

2.2 The Shuffle Model

In the shuffle model, users add noise locally but then a shuffler randomly permutes the values before they get published to the analyst. Intuitively, the idea is that users can add less noise compared to the *local model* and the noise created by the shuffling will compensate for the difference.

Let n users with each user i holding a secret x_i. A mechanism (R, S, A) in the shuffle model consists of:

- **Local Randomizer** $R : \mathcal{X} \to \mathcal{Y}^*$. A randomized encoder used by each user i: takes as input x_i and outputs a vector y_i with length m . If $m = 1$, we define it as the *one-message shuffle model* and if $m > 1$ as the *multi-message shuffle model* .
- **Shuffler** $S : \mathcal{Y}^* \to \mathcal{Y}^*$. Takes as input the vectors $y_1, \ldots, y_n$, each with length m, and shuffles their bits altogether. In other words, it first concatenates them to a single bit vector, creating $y_{1,1}, \ldots, y_{n,m}$, then chooses a uniformly random permutation $\pi : \mathbb{N} \to \mathbb{N}$ and finally outputs $y_{\pi(1,1)}, \ldots, y_{\pi(n,m)}$. We assume that only a "primitive" shuffler can be used; we further discuss this in Section 8.
- **Analyst** [1] $A : \mathcal{Y}^* \to \mathbb{R}$. Uses an *analysis function* that takes as input a set of messages $y_{1,1}, \ldots, y_{n,m}$ and tries to estimate a function $f(x_1, \ldots, x_n)$ from these messages.

The result of shuffling the locally randomized values can be viewed as a mechanism: $M(X) = S(R(x_1), \ldots, R(x_n))$. Hence we can adapt the definition of Differential Privacy in this model as follows:

Definition 3 (**Shuffle model of Differential Privacy**). *[11, 18] A mechanism (R, S, A) is (ε, δ)- approximate differentially private iff $M(X) = S(R(x_1), \ldots, R(x_n))$ is (ε, δ)- approximate differentially private.*

[1] We recall here the definition of the mechanism given in [11, 18], which includes an entity called the "analyst" representing the final use of the output. This is to underline the fact that, in general, a mechanism is designed for a specific notion of utility.

2.3 Metric Privacy

The notion of distance used in Differential Privacy, namely the Hamming distance, depends only on the number of different records between the two datasets, without considering *how much* different they actually are. In other words standard Differential Privacy aims at fully hiding the value of each individual; any change in the value is equally hidden.

Metric privacy, a generalisation of Differential Privacy, aims at addressing this limitation, by considering an arbitrary set of secrets $\mathcal{X}$, equipped with a *metric* d_x, where $d_x(x, x')$ denotes how indistinguishable x and x' are required to be. Then, metric privacy requires that, the closer (wrt d_x) two secrets are, the more similar the outcomes of the mechanism on these secrets should be.

Definition 4 (**Approximate Metric Privacy**). [2] *A mechanism M satisfies approximate (ε, δ) - d_x - privacy, iff for all secrets x, $x' \in \mathcal{X}$ and sets of results S:*

$$\mathbb{P}[M(x) \in S] \leq e^{\varepsilon \cdot d_x(x,x')} \, \mathbb{P}[M(x') \in S] + \delta$$

In the subsequent sections of this work, in order to express that a mechanism satisfies approximate metric privacy, we will simply write that is satisfies (ε, δ) - d_x - privacy. Moreover if $\delta = 0$ we say that the mechanism satisfies pure d_x - privacy.

Note that the use of an arbitrary set of secrets $\mathcal{X}$ in the above definition, with its corresponding metric d_x, allows us to adapt metric privacy to the various privacy models, by selecting $\mathcal{X}$ and d_x accordingly:

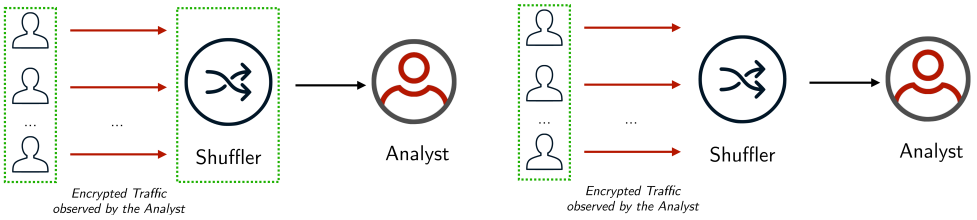
- *Local model.* Here $\mathcal{X}$ is the domain of user values $\mathcal{D}$. In this paper we consider numeric data, hence the domain is $\mathbb{R}$ and d_x is the Euclidean distance, which will be denoted as $d_{\mathbb{R}}$. In the case of location data, the domain is $\mathbb{R}^2$ and the metric is $|| \cdot ||_2$.
- *Central model.* Here $\mathcal{X}$ is the set of all datasets. As for the metric, in this paper we consider the Manhattan distance, i.e.:
$$d_X(X, X') = \textstyle\sum_i d_x(x_i, x'_i) \,, \tag{1}$$
where d_x is an underlying metric on the values (in our case the Euclidean distance). Hence, d_X measures the total change in the values between the two datasets, instead of just the number of modified records.
- *Shuffle model.* Similarly to the central model, here $\mathcal{X}$ is the set of all datasets with the metric d_X from (1). The only difference is that the employed mechanism $M(X) = S(R(x_1), \ldots, R(x_n))$ involves a Local Randomizer R and a shuffler S.

We notate with d the result of d_X. The most widely used application of metric privacy is the case when $\mathcal{X} \subseteq \mathbb{R}^2$ are locations and d_x is the Euclidean distance, known in the literature as *geo-indistinguishability* [1].

Finally, note that in the central model, we can achieve metric privacy by simply applying a local mechanism to the whole dataset X. If $M(x)$ satisfies (ε, δ)-d_x-privacy, then the mechanism $M(X) = (M(x_1), \ldots, M(x_n))$ satisfies (ε, δ)-d_X-privacy. However, if we only wish to answer a specific numeric query $f(X)$, we can achieve

[2] A weaker variant of this definition has been proposed in [2], requiring that $\mathbb{P}[M(x) \in S] \leq e^{\varepsilon d_x(x,x')} \mathbb{P}[M(x') \in S] + \delta e^{d_x(x,x')}$, meaning that δ increases with the distance (similarly to ε). For this work, the stronger variant is sufficient, the results however can be directly translated to the weaker variant.

Figure 1: Trust Boundaries (in green)



(a) when the shuffler is trusted **(b) when the shuffler is corrupted**

stronger privacy in the central model by applying M only to the outcome of f, instead of the data, and adapting the noise to the *sensitivity* of f as follows:

Theorem 2.2. *[1] Let $f(X)$ be a numeric query on a dataset X and let Δ_f be its* sensitivity*, defined as:*

$$\Delta_f = \max_{X,X'} \frac{d_{\mathbb{R}}(f(X), f(X'))}{d_X(X, X')} .$$

If a mechanism $M(x)$ satisfies (ε, δ)-$d_{\mathbb{R}}$-privacy then $M(f(X))$ satisfies $(\Delta_f \cdot \varepsilon, \delta)$-$d_X$-privacy.

2.4 Adversary Model

In this work we consider the analyst to be the adversary who receives the output of the shuffler and tries to determine the value of a user. First, we follow the traditional assumption of the shuffle model and assume that the shuffler is trusted and not controlled by the analyst (Figure 1a). Note that, in this case, the privacy of the proposed mechanisms only depends on the output of the shuffler and not on any extra actions possibly performed by the analyst (Lemma 2.1). This is to emphasize that the role of the analyst solely impacts the utility of the mechanism.

Then, in Section 7 we continue by challenging the assumption that the shuffler is trusted and examine the privacy of the proposed protocols when the analyst corrupts the shuffler (Figure 1b).

Furthermore, in both cases, we consider that the adversary can monitor the communication channel between the users and the shuffler. Although the messages are encrypted, their metadata could reveal information. This necessitates the requirement for every user to send messages of the same length (further discussed in Section 8).

Finally, we assume that the adversary does not control any users. We expect all users to behave in at least an honest-but-curious way, meaning that each will properly execute their Local Randomizer and send their obfuscated value to the shuffler.

3 Randomized Response in the Shuffle Model

The core mechanism . First let us describe Randomized Response (RR) [31]. Consider a user in the local model with a value x from a binary domain $\mathcal{D} = \{0, 1\}$. RR performs a probabilistic choice: with probability p (a parameter of the mechanism) it ignores x and uniformly reports a value in $\{0, 1\}$ and with probability $1 - p$ it reports x truthfully.

Randomized Response can also be used in the case of a non-binary domain $\mathcal{D}$; in this case it is usually called k-RR, where $k = |\mathcal{D}|$. Similarly to the binary case, the mechanism reports a random value from $\mathcal{D}$ with probability p and x with probability $1 - p$. Note that k-RR treats all values that are different from x in the same way, independently from their distance from x. In other words, when $x = 0$, the probability of reporting 1 or 10^{10} is the same. However, recall that, in metric privacy, a mechanism should report something close (wrt a metric function d_x) to the initial value. This makes k-RR a poor choice for metric privacy.

3.1 RR for Standard DP in the Shuffle Model

The idea of improving the privacy (for standard DP) of RR with a shuffler was introduced in [11]. The author first discussed *Binary RR-Shuffle*, a protocol for binary summation where each user i with a secret x_i runs $\text{RR}(x_i, p)$ and then sends their obfuscated bit to the shuffler. After receiving all the bits, the shuffler randomly permutates them and releases the output to the analyst who may sum it (and debias it to get a more accurate result).

For real summation an encoding step is necessary, where the real value of each user i is encoded to a bit vector (b_i) of size r. Then *Binary RR-Shuffle* can be run for each bit of the vector. This allows us to estimate the sum $\sum_i b_i^j$ of the j-th bit of all users. The analyst may then add them together and divide by r to compute the sum of all x_i's.

3.2 Adapting RR for Metric Privacy in the Shuffle Model

We now turn our attention to the problem of reporting the sum of values from a discrete domain $\mathcal{D} = \{0, \ldots, k\}$ while satisfying *metric privacy* in the shuffle model. Since k-RR does not respect the underlying metric of the values, we employ a variation of *Binary RR-Shuffle*, using a unary encoding inspired by the real summation protocol of [11]. Each value $x_i \in \mathcal{D}$ is encoded as a bit vector $(b_i^1, \ldots, b_i^k)$ with *exactly* x_i ones. Then, RR is applied to each bit.

Algorithm 1: $\mathcal{U}(x, r)$: Unary encoding of x in r bits

Input : $x \in \mathbb{N}, r \in \mathbb{N}$, where $x \leq r$
Output: $(b_1, \ldots, b_r) \in \{0, 1\}^r$
$b \leftarrow \{0\}^r$
for $j = 1, \ldots, x$ **do**
  $b_j = 1$
end
Return $(b_1, \ldots, b_r)$

The protocol *RR-Shuffle*(δ, p, k, n) works as follows:

Local Randomizer:

- Each user i encodes his secret x_i to a bit vector b_i using $\mathcal{U}(x_i, k)$.
- Then, for each bit j of b_i they run RR(j, p), creating the obfuscated bit vector b'_i which is sent to the shuffler.

Shuffler:

- The shuffler receives all the bit vectors, concatenates them into a single bit vector, randomly permutes it, creating the permutated bit vector B which is finally released to the analyst.

Analyst:

- The analyst may sum B and perform a debiasing to increase the utility of the mechanism. Let $\lambda = p \cdot n \cdot k$. The debiased result z is computed as:

$$z = \frac{n \cdot k}{n \cdot k - \lambda} \cdot \left(\sum_{i=1}^{n \cdot k} B_i - \frac{\lambda}{2} \right)$$

Note that, in contrast to k-RR, this procedure respects the underlying (i.e. Euclidean) metric. Noise is applied to each bit of x_i independently, and the noisy bits are added. This causes values close to x_i to be produced with higher probability than values further away from x_i, since the latter requires a larger number of bits to be flipped.

Algorithm 2: RR-Shuffle(δ, p, k, n)

Input: n users, privacy parameter δ, maximum possible value k, parameter of RR p.
Output: debiased result z

Local Randomizer (each user i):
$b_i \leftarrow \mathcal{U}(x_i, k)$
$b'_i \leftarrow \{RR(b_{i,0}, p), \ldots, RR(b_{i,k}, p)\}$

Shuffler:
Collect all bit vectors b'_i from users.
$B \leftarrow \{b'_{0,0}, \ldots, b'_{n,k}\}$
Send $\{B_{\pi(0)}, \ldots, B_{\pi(n \cdot k)}\}$ to the analyst.

Analyst:
Receive the permutated vector B from the shuffler.
$\lambda \leftarrow p \cdot n \cdot k$
$z = \frac{n \cdot k}{n \cdot k - \lambda} \cdot \left(\sum_{i=1}^{n \cdot k} B_i - \frac{\lambda}{2} \right)$
Return z

A natural question arises: why does the user need to send their noisy value as a vector of bits rather than an integer? We are going to answer this question in detail in Section 4.1.1, but a high-level argument follows. Observe that, in the integer case, the outliers would still be recognizable after shuffling, albeit with some noise. By converting the values into bits and shuffling them, the adversary cannot distinguish an outlier. For example, he cannot determine if one user has a particularly large value (x_L) or n_s users have smaller values s.t. $\sum_{i=1}^{n_s} x_i = x_L$; in both cases the mechanism produces the same output. Consequently, with this unary conversion, the amount of information that the adversary obtains is significantly reduced.

We can then show that RR-Shuffle satisfies (ε, δ) approximate metric privacy:

Theorem 3.1. *For any $\delta > 0$,$0 \leq p \leq 1$ and $n \cdot k \geq \lambda \geq 14 log \frac{4}{\delta}$, let $\lambda = p \cdot n \cdot k$.*
RR-Shuffle(δ, p, k, n) is (ε, δ) - d_X - private where

$$\varepsilon = \sqrt{\frac{32 \log \frac{4}{\delta}}{\lambda - \sqrt{2\lambda \log \frac{2}{\delta}}}}$$

The proof can be found in Appendix A. Finally, observe that each user has to send k bits but this is further discussed in Section 8.

3.3 Utility of RR-Shuffle

Following the approach of [11] we can present a utility bound for the RR-Shuffle mechanism:

Theorem 3.2. *Let P be RR-Shuffle(δ, p, k, n) and $\lambda = p \cdot n \cdot k$. For every $n \in \mathbb{N}, 0 < \beta \leq 1, nk > \lambda \geq 2 log \frac{2}{\beta}$ and $X = \{0,1\}^{nk}$:*

$$\mathbb{P}\left[\left| P(X) - \textstyle\sum_i x_i \right| > \frac{nk}{nk - \lambda} \sqrt{2\lambda \log \frac{2}{\beta}} \right] < \beta$$

The proof can be found in Appendix A.

4 Geometric Mechanism in the Shuffle Model

In Section 3.2 we showed that by using a unary encoding, we can achieve metric privacy via Randomized Response in the shuffle model, although RR itself does not respect the underlying metric. Nevertheless, the natural question that arises is whether we can apply shuffling to a mechanism that does respect the underlying metric, since one would expect such a mechanism to be better suitable for metric privacy.

This is the goal of this section, in which we study the *geometric mechanism* (aka *discrete Laplace mechanism*), a natural choice to achieve metric privacy.

The core mechanism. The geometric mechanism [23] draws noise from the double geometric distribution, a discrete version of the Laplace distribution with a parameter ε_{geo}. In the local model, the geometric mechanism $G(x, \varepsilon_{\text{geo}})$, applied to the user's value x, produces a value y with probability decreasing exponentially with the distance between x and y, as follows:

$$\mathbb{P}[G(x, \varepsilon_{\text{geo}}) = y] = \frac{1 - e^{-\varepsilon_{\text{geo}}}}{1 + e^{-\varepsilon_{\text{geo}}}} \cdot e^{-\varepsilon_{\text{geo}}|x-y|} \tag{2}$$

From (2) we can easily conclude that the geometric mechanism satisfies ε_{geo}-$d_{\mathbb{R}}$-privacy.

We can also use the geometric mechanism in the central model to answer a query $f(X)$, by adapting its noise to the sensitivity of f, as discussed in Section 2.3. For instance, the *average* query on a dataset of n users has sensitivity $\Delta_f = \frac{1}{n}$, so the central-model mechanism $G(f(X))$ satisfies $\frac{\varepsilon_{\text{geo}}}{n}$-$d_X$-privacy. In other words, in the central model we achieve much stronger privacy with the same noise, since the noise is only applied to the result of a low-sensitive query.

In the following, we will refer to the application of the geometric mechanism in the local and central models respectively as *Geo-Local* and *Geo-Central*.

4.1 Enhancing the Geometric Mechanism with a Shuffler

We are now ready to discuss how the privacy of Geo-Local can be improved via the use of a shuffler. The goal is to apply G (the geometric mechanism) with parameter ε_{geo}, but exploit the boost of the shuffler to obtain stronger than ε_{geo}-d_X-privacy. In the remainder of this section, we will use ε to indicate the privacy loss of the Geo-Shuffle mechanism and ε_{geo} for the privacy loss of Geo-Local.

Issues with the direct approach. First, let us combine the geometric mechanism with a shuffler in a direct way: we apply local noise to each value, obtaining from each user i the value $y_i = G(x_i, \varepsilon_{\text{geo}})$, then we apply the shuffler to $(y_1, \ldots, y_n)$ and publish the shuffled vector. Although this straightforward approach seems natural, it fails to provide stronger than ε_{geo} metric privacy. To understand why, consider a dataset in which x_1 is a large value, say 10^{10}, and all other values are 0. Since the geometric mechanism produces values close to the original one with higher probability, y_1 will almost surely be much larger than the remaining y_i's. As a consequence, an adversary can distinguish which of the shuffled values is y_1, defeating the effect of the shuffler. This means that, in order for the shuffler to provide an actual boost, we need to effectively hide the relationship between the original and the shuffled values.

Overcoming the hurdle. We are going to exploit once again the unary encoding step used in Section 3.2. The general idea is the following: we first apply the geometric mechanism, then we truncate the outcome within a bounded interval and produce a unary encoding of the obtained bounded value. Finally, the shuffler permutes these encoded values and releases the output.

Let n users, with each user i holding x_i, an integer value in $\{0, \ldots, k\}$ for a $k \in \mathbb{N}$. *Geo-Shuffle*$(\varepsilon_{\text{geo}}, \delta, n)$ works as follows:

Local Randomizer:

- Each user i runs Geo-Local with parameter ε_{geo} to sample the noise N_i, e.g. $N_i \sim G(0, \varepsilon_{\text{geo}})$.
- Then, they compute x_i':

$$x_i' = \begin{cases} x_i + N_i + c & 0 \le x_i + N_i + c \le k + 2c \\ 0 & x_i + N_i + c < 0 \quad \text{(truncate to 0)} \\ k + 2c & x_i + N_i + c > k + 2c \quad \text{(truncate to k+2c)} \end{cases} \tag{3}$$

where $c = \left\lceil - \frac{\ln\left[\frac{\left(1+e^{-\varepsilon geo}\right)\left(1-\sqrt[n]{1-\frac{\delta}{2}}\right)}{2}\right]}{\varepsilon_{geo}} \right\rceil$

- Finally, they encode x_i' to a bit vector b_i using $\mathcal{U}(x_i', k + 2c)$ and send it to the shuffler.

Shuffler:

- The shuffler receives all the bit vectors, concatenates them into a single bit vector, randomly permutes it and then releases it to the analyst.

Analyst:

- The analyst may sum the permutated bit vector. They can perform a debiasing to increase the utility of the mechanism by subtracting $n \cdot c$.

The Geo-Shuffle mechanism is a multi-message shuffle protocol. Each user has to send $k + 2c$ bits but this is further discussed in Section 8.

Algorithm 3: `Geo-Shuffle`$(\varepsilon_{\text{geo}}, \delta, n)$

Input: n users, privacy parameters $\varepsilon_{\text{geo}}, \delta$
Output: debiased result z

$c \leftarrow \left\lceil - \frac{\ln\left[\frac{\left(1+e^{-\varepsilon geo}\right)\left(1-\sqrt[n]{1-\frac{\delta}{2}}\right)}{2}\right]}{\varepsilon_{geo}} \right\rceil$

Local Randomizer (each user i):

$N_i \sim G(0, \varepsilon_{\text{geo}})$

$$x_i' \leftarrow \begin{cases} x_i + N_i + c, & 0 \le x_i + N_i + c \le k + 2c \\ 0, & x_i + N_i + c < 0 \\ k + 2c, & x_i + N_i + c > k + 2c \end{cases}$$

Send $\mathcal{U}(x_i', k + 2c)$ to the shuffler.

Shuffler:

Collect all bit vectors b_i from users.
$B \leftarrow \{b_{0,0}, \ldots, b_{n,k+2c}\}$
Send $\{B_{\pi(0)}, \ldots, B_{\pi(n\cdot(k+2+c))}\}$ to the analyst.

Analyst:

Receive the permutated vector B from the shuffler.
$z \leftarrow \sum_i B_i - n \cdot c$
Return z

4.1.1 Privacy analysis. We provide here an overview of the privacy analysis of Geo-Shuffle, referring to the *Section B* for the full proofs. Note that, unlike RR-Shuffle, the privacy analysis of Geo-Shuffle is not subject to parameter restrictions.

Main idea. The core substance is the unary representation which, as we are about to explore, overcomes the limitations of the direct approach and protects the outliers. This is because releasing a shuffled bit vector is privacy-wise equivalent to releasing its sum:

Proposition 1. *Let a mechanism $K(X) = \sum_{x \in X} R(x)$ be (ε, δ) - d_X - private for a dataset X and R be a Local Randomizer. Then, the shuffle model mechanism with a shuffler S (as defined in Section 2.2) which uses the unary encoding $\mathcal{U}(x, r)$ (Algorithm 1):*

$$M(X) = S\Big(\{\mathcal{U}(R(x), r) : x \in X\}\Big)$$

is also (ε, δ) - d_X - private.

For instance, consider a bit vector of size 5 that contains 3 ones and 2 zeros. Then, the following statements give exactly the same level of information to the adversary: *"the sum of the vector is 3"*, *"the values of the shuffled vector are* $\{0, 0, 1, 1, 1\}$*"*. We need to stress that this is only the case when the vector has unary values. For example, consider an integer vector with values $\{0, 5, 10\}$. Clearly now these statements are not privacy-wise equivalent: *"the sum of the vector is 15"* ,*"the values of the shuffled vector are* $\{0, 5, 10\}$*"*. This example clarifies why we could not directly use Geo-local and shuffle all its outputs.

It is important to avoid a confusion between the summation of the outputs of the Local Randomizers, which will be used for reasoning about privacy (e.g. K in Proposition 1) and the summation performed by the analyst. The output of the mechanism is the output of the shuffler; any operation thereafter (for example the analyst summing all the shuffled bits) is considered post-processing.

In other words, whatever the analyst does with the shuffled bits does not affect the privacy of this mechanism, which always equates to disclosing the sum of the bits.

Achieving a unary representation. First, let us explain the motivation behind the truncation happening in Equation (3). If users just submitted $x_i + N_i$ (their value plus noise), then unary encoding might not be always possible (using Algorithm 1), since N_i can be negative. To solve this problem, we *shift by* $c \in \mathbb{N}$ the user's obfuscated value. If the final result (initial secret + noise + c) is outside of $[0, k + 2c]$, then it is truncated to the nearest bound.

Now we need to bound the probability of this truncation happening, as it will be necessary for the rest of the proof. To do this, we study the probability that at least one of n users samples a noise that is $\notin [-c, c]$. Due to the fact that the geometric mechanism will produce a value close to the initial one with high probability, for a reasonable choice of ε_{geo}, the following proposition dictates that this probability is small, and hence can be covered by $\delta/2$. We show that:

Proposition 2. *Let $N_i \sim G(0, \varepsilon_{\text{geo}})$, $\delta \in (0, 1]$ and :*

$$c = \left\lceil -\frac{\ln\left[\frac{\left(1+e^{-\varepsilon_{geo}}\right)\left(1-\sqrt[n]{1-\frac{\delta}{2}}\right)}{2}\right]}{\varepsilon_{geo}} \right\rceil$$

Then:

$$1 - \prod_{i=1}^{n} \mathbb{P}\left[N_i \in [-c, c]\right] \leq \frac{\delta}{2}$$

Observe that we did not take into account the input dataset. To put it simply, we did not consider that a user having a value $x = k$ is more likely to truncate than someone having a value $x' = k/2$. In other words, this is not just the probability that every user picks the first branch Equation (3), which of course depends on the input dataset. This is instead a sub-event of picking the first branch: the amount of noise is so small that users always select the first branch regardless of their x_i. In other words, it is not that $x_i + N_i$ is bounded, but that N_i alone is bounded, considering the worst case scenarios for x_i (i.e. $x_i = k$ for the upper bound and $x_i = 0$ for the lower bound). This enabled us to find an upper bound regardless of the input dataset, leading to the following corollary:

Corollary 1. *If n users run Geo-Shuffle($\varepsilon_{\text{geo}}, \delta, n$), each with a secret x_i, the probability that at least one of them reports a truncated x_i' is at most $\delta/2$, regardless of each x_i.*

Studying the case when nobody truncates. Next, we show that it suffices to study the privacy of Geo-Shuffle when no user has to truncate (i.e. every x_i' comes from the first branch of Equation (3) because N_i is small). If we prove that in that case, the mechanism is $(\varepsilon, \frac{\delta}{2})$-metric private, then, using Proposition 3, we will be able to conclude that Geo-Shuffle is (ε, δ)-metric private, in every case (someone submitting a truncated value or not).

Proposition 3. *Let M be a mechanism and H be an event independent of the input of M such that $\Pr[H] \geq 1 - \delta_1$. Moreover, assume that when H occurs M is (ϵ, δ_2)- d_X - private, namely for all X, X':*

$$\Pr[M(X) \in S|H] \leq e^{\epsilon \cdot d_X(X,X')} \Pr[M(X') \in S|H] + \delta_2$$

If K is a mechanism that behaves like M when H occurs then K is $(\epsilon, \delta_1 + \delta_2)$ - d_X - private.

Here we apply Proposition 3 by setting H as the event that nobody truncates, which is irrelevant of the input dataset (Corollary 1), K as the Geo-Shuffle mechanism and M as the special case of Geo-Shuffle when nobody truncates (because N_i is small, regardless of their x_i), which we are about to study its privacy. Regarding δ we set $\delta_1 = \delta/2$ (Proposition 2) and $\delta_2 = \delta/2$.

Geo-Shuffle when nobody truncates. Now let us examine the scenario in which no user truncates their obfuscated value. Let us notate the total noise generated by all the users, in this scenario, by $N_{total} = \sum_i N_i$.

First, we prove that Geo-Local can be described by the difference between two geometric distributions. Successively, this can be expressed as the difference between two Negative Binomial distributions (NB): $N_{total} = NB(n, 1 - e^{-\varepsilon_{\text{geo}}}) - NB(n, 1 - e^{-\varepsilon_{\text{geo}}})$.

In turn, this difference has been studied in the literature as the *symmetric* version of the *Generalized Discrete Laplace distribution* [28] (SGDL) i.e.: $N_{total} = SGDL(n, e^{-\varepsilon_{\text{geo}}})$. As far as we are aware, SGDL has not been studied before in DP or metric privacy.

The SGDL distribution has the following PMF, for Y ~ SGDL (β, p) and any $r \in \mathbb{Z}$:

$$\mathbb{P}(Y = r) = (1-p)^{2\beta} \sum_{k=|r|}^{\infty} \binom{\beta + k - 1}{k} \binom{\beta + k - |r| - 1}{k - |r|} p^k p^{k-|r|}$$

Let us now briefly present this distribution, visualized in Figure 2 for multiple values of β, p. SGDL(β, p) represents the difference between two Negative Binomial Distributions (NB) with parameters $\beta, 1 - p$. In Geo-Shuffle's case, the parameters are $\beta = n$ and $p = e^{-\varepsilon_{\text{geo}}}$.

By studying the SGDL distribution we prove that this special case of Geo-Shuffle (when nobody has to truncate) is $(\varepsilon, \frac{\delta}{2})$ metric private with:

$$\varepsilon = \max_{d_X} \frac{ln(g(-\alpha, d_X(X, X'), e^{-\varepsilon_{\text{geo}}}, n))}{d_X(X, X')} \tag{4}$$

for a function g and an α as defined in Theorem 4.1.

Note that after conducting numerous experiments with every possible combination of parameters (presented in Appendix B), we have observed that the maximum ε results from $d_X = 1$. Finally,

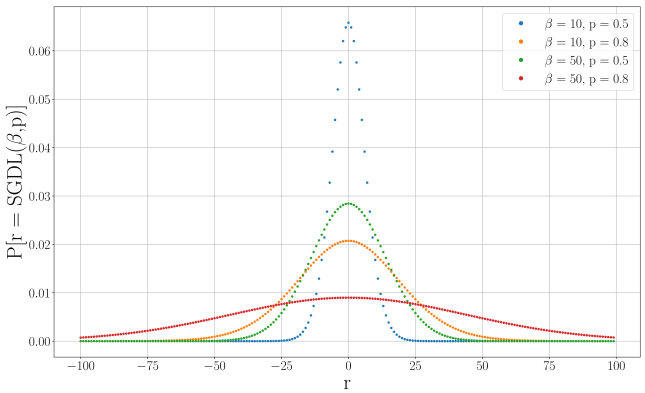


Figure 2: PMF of SGDL

using Proposition 3 we can conclude about the privacy of Geo-Shuffle in every scenario (regardless of someone truncating or not):

Theorem 4.1. *Let $\delta \in (0,1]$. The Geo-Shuffle($\varepsilon_{\text{geo}}, \delta, n$) mechanism is (ε,δ)- d_X - private, with:*

$$\varepsilon = \max_{d_X} \frac{ln(g(-\alpha, d_X(X,X'), e^{-\varepsilon_{\text{geo}}}, n))}{d_X(X,X')} \tag{5}$$

where

$$g(r,d,e^{-\varepsilon_{\text{geo}}},n) = \max\{h(r,d,e^{-\varepsilon_{\text{geo}}},n), \frac{1}{h(r,d,e^{-\varepsilon_{\text{geo}}},n)}\}$$

$$h(r,d,p,n) = \frac{\sum_{k=\lfloor r-\frac{d}{2}\rfloor}^{\infty} \binom{n+k-1}{k}\binom{n+k-\lfloor r-\frac{d}{2}\rfloor-1}{k-\lfloor r-\frac{d}{2}\rfloor} p^k p^{k-\lfloor r-\frac{d}{2}\rfloor})}{\sum_{k=\lfloor r+\frac{d}{2}\rfloor}^{\infty} \binom{n+k-1}{k}\binom{n+k-\lfloor r+\frac{d}{2}\rfloor-1}{k-\lfloor r+\frac{d}{2}\rfloor} p^k p^{k-\lfloor r+\frac{d}{2}\rfloor}}$$

$$t = 2\cdot\frac{\varepsilon_{\text{geo}}}{\sqrt{n}} \qquad p = e^{-\varepsilon_{\text{geo}}}$$

$$\alpha = -\frac{\ln(\frac{\delta}{4}\cdot[\frac{(1-p)^2}{(1-pe^t)(1-pe^{-t})}]^{-n})}{t}$$

4.2 Privacy Boost

In the previous section we showed why Geo-Shuffle can achieve (ε,δ) approximate metric privacy using Geo-Local as a Local Randomizer which is ε_{geo} metric private. Informally, consider that we used a mechanism with ε_{geo} privacy and the extra noise created by shuffling resulted in ε privacy. The crucial observation that has to be made is that $\varepsilon < \varepsilon_{\text{geo}}$. As more users use Geo-Shuffle, ε further decreases, because, intuitively, shuffling more values creates more noise.

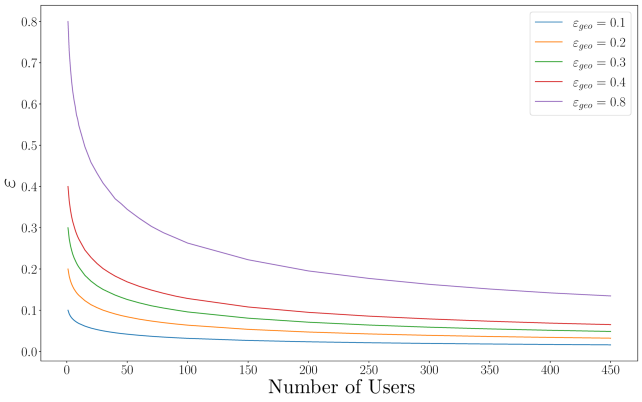


Figure 3: Privacy boost of Geo-Shuffle, ε and ε_{geo} are its privacy levels in the shuffle and local model respectively.

The complex formula of Theorem 4.1 does not make that observation clear. To clarify, we visualize this privacy boost in Figure 3. For only one user there is no privacy boost, because essentially both mechanisms are the same. But even for a few users ε is remarkable smaller. As the number of users increases, the shuffling effect is amplified, offering even better privacy. Equivalently, one might consider that for the same level of privacy, Geo-Shuffle can produce less noisy results, offering better utility, as we experimentally show in Section 6.

5 SGDL in the Shuffle Model

In this section, we provide a mechanism that has with high probability optimal accuracy wrt the central model. Our motivation is the Correlated Noise protocol suggested by Badih Ghazi et.al. [21] for standard DP. We begin by discussing the original mechanism, then we describe our adaptation in metric privacy.

5.1 The Correlated Noise Mechanism [21]

Recall that summing bits releases the same level of information as shuffling them (Proposition 1). The core idea of the protocol of [21] is that users add noise s.t. the sum of all these noises is equal to the noise produced by the Geo-Central mechanism. Hence we need to deconstruct Geo-Central into a sum of n i.i.d. samples, one for each user. Formally, this is described by the notion of *infinite divisibility*. The geometric mechanism is infinitely divisible [25] and this property has already been used in standard DP [5] [21]. Specifically, to achieve this, each user has to add noise sampling from $SGDL(1/n, e^{-\varepsilon}) = NB(1/n, 1-e^{-\varepsilon}) - NB(1/n, 1-e^{-\varepsilon})$ [3]. The first NB distribution can be considered as the positive noise and the second as the negative noise.

The original mechanism of [21] involves each user submitting two messages: one with their secret along with the positive noise (first NB distribution) and one with only the negative noise (second NB distribution). Since the analyst observes each type of message separately, they could greatly reduce the amount of noise simply by dropping the negative messages. To prevent this, 0.01ε is allocated to a third mechanism (called *correlated noise*) to add some extra noise to both messages. Each user samples this noise once and adds it to both messages, hence the utility of the mechanism is not affected as adding all messages together cancels out this extra noise. However, the adversary can no longer deduce as much information just by examining the positive messages.

Our adaptation. We adapt this mechanism to metric privacy, proposing *SGDL-Shuffle* by making the following changes. First we again use the SGDL distribution to describe the difference between two NB distributions, to stay consistent with our analysis of Geo-Shuffle (Section 4). This will also be advantageous when studying the privacy guarantees of the mechanism in the event that the shuffler is compromised (Section 7). Moreover, we refrain from using the correlated noise approach, to be consistent with our requirement that every user sends the same number of bits. Instead, we use the same approach of *shifting by c*, as we did in Section 4. This leads us to spend a small amount of δ, but in turn we will avoid spending 0.01ε, making our approach simpler.

5.2 The SGDL-Shuffle Mechanism

In this section we describe our proposed mechanism for metric privacy in detail. Consider n users, with each user i holding x_i, an integer value in $\{0,\dots,k\}$ for a $k \in \mathbb{N}$.

Let $u = 1 + \frac{\sqrt{w(w-4)}-w}{2}$, where $w = 2n\cdot\ln\left(\frac{1-\sqrt[n]{1-\delta}}{2}\right)$

The *SGDL-Shuffle(ε,δ,n)* protocol works as follows:

Local Randomizer:

[3] A similar technique of noise division has been studied in distributional DP [29, 33].

- Each user i samples noise N_i from the Symmetric Generalized Discrete Laplace distribution: $N_i \sim SGDL(\frac{1}{n}, e^{-\varepsilon})$.
- Then, they compute x'_i:

$$x'_i = \begin{cases} x_i + N_i + c & 0 \le x_i + N_i + c \le k + 2c \\ 0 & x_i + N_i + c < 0 \quad \text{(truncate to 0)} \\ k + 2c & x_i + N_i + c > k + 2c \quad \text{(truncate to k+2c)} \end{cases} \tag{6}$$

  where: $c = \left\lceil \frac{u(1-e^{-\varepsilon})}{n} \right\rceil$
- Finally, they encode x'_i to a bit vector b_i using $\mathcal{U}(x'_i, k + 2c)$ and send it to the shuffler.

Shuffler:

- The shuffler receives all the bit vectors, concatenates them into a single bit vector, randomly permutes it and then releases it to the analyst.

Analyst:

- The analyst may sum the permutated bit vector. They can perform a debiasing to increase the utility of the mechanism by subtracting $n \cdot c$.

The SGDL-Shuffle mechanism is a multi-message shuffle protocol. Each user has to send $k + 2c$ bits where $c = f(\varepsilon, \delta, n)$.

Algorithm 4: `SGDL-Shuffle`(ε, δ, n)

Input: n users, privacy parameters ε, δ
Output: debiased result z

$w \leftarrow 2n \cdot \ln\left(\frac{1-\sqrt[n]{1-\delta}}{2}\right)$
$u \leftarrow 1 + \frac{\sqrt{w(w-4)}-w}{2}$
$c \leftarrow \left\lceil \frac{u(1-e^{-\varepsilon})}{n} \right\rceil$
Local Randomizer (each user i):
$N_i \sim SGDL(\frac{1}{n}, e^{-\varepsilon})$

$$x'_i = \begin{cases} x_i + N_i + c & 0 \le x_i + N_i + c \le k + 2c \\ 0 & x_i + N_i + c < 0 \\ k + 2c & x_i + N_i + c > k + 2c \end{cases}$$

Send $\mathcal{U}(x'_i, k + 2c)$ to the shuffler.

Shuffler:
Collect all bit vectors b_i from users.
$B \leftarrow \{b_{0,0}, \ldots, b_{n,k+2c}\}$
Send $\{B_{\pi(0)}, \ldots, B_{\pi(n\cdot(k+2+c))}\}$ to the analyst.

Analyst:
Receive the permutated vector B from the shuffler.
$z \leftarrow \sum_i B_i - n \cdot c$
Return z

In the following paragraphs, we outline the privacy and utility of SGDL-Shuffle, referring readers to Section C for the full proofs.

Privacy Analysis. Following Section 5.1, observe that the protocol would have had the same output as the geometric mechanism in the central model, if the users never truncated their noise; in other words, always choosing the first case of Equation (6). We show that this happens with probability $1 - \delta$ (Lemma C.1), regardless of the input dataset, using the fact that SGDL will return a value close to 0 with high probability (similarly to Section 4.1). In other words, with probability $1 - \delta$ these two mechanisms are identical; this enables to employ again Proposition 3 to prove the privacy of SGDL-Shuffle by setting H as the event that nobody truncates (because each N_i is small, regardless of each x_i), K as the SGDL-Shuffle mechanism and M as the special case of SGDL-Shuffle where nobody truncates. We set $\delta_1 = \delta$ and $\delta_2 = 0$. This leads us to the following theorem:

Theorem 5.1. *For any $\delta \in (0, 1]$ SGDL-Shuffle (ε, δ, n) is (ε, δ) - d_X - private.*

Observe that, similar to Geo-Shuffle but unlike RR-Shuffle, there are no parameter restrictions.

Utility Analysis. If all users choose the first case of Equation (6), the analyst would see the same output as the Geo-Central mechanism, which has been shown to be universally optimal [9, 23]. We showed that this happens with probability $1 - \delta$, which is negligible because typically δ is small. Thus, we can immediately conclude in Proposition 4; the utility of SGDL-Shuffle is with probability $1 - \delta$ the same as that of Geo-Central.

Proposition 4. *For any integer dataset X with n users and any $\delta \in (0, 1]$, if $A \sim$ SGDL-Shuffle$(X, \varepsilon, \delta, n)$ and $B \sim$ Geo-Central(X, ε), then A and B follow the same distribution w.p. $1 - \delta$.*

6 Experimental Evaluation

In this section, we experimentally evaluate the utility of the proposed shuffle mechanisms for metric privacy, namely *RR-Shuffle*, *Geo-Shuffle* and *SGDL-Shuffle*. We compare the mechanisms to each other, but also, as a baseline, to *Geo-Local*, the standard geometric mechanism in the local model. The mechanisms are evaluated on two datasets: one with synthetic numeric data and one with real-world location data. In the following sections, we first describe the experimental setup and then present the results for both datasets. We present the most notable results for each experiment, directing readers to Section E for additional experiments with varying parameters and utility metrics.

Summary of Findings. A summary of findings is shown in Table 2 and Table 3 for the first and second experiment respectively, showing the utility loss of each mechanism for multiple parameters. We can observe that SGDL-Shuffle greatly outperforms all other mechanisms, scoring near-optimal utility. The second-best mechanism is Geo-Shuffle, offering comparable results as the number of users (n) increases. Although RR-Shuffle performs worse than both of them, it still offers, most of the time, better utility compared to Geo-Local.

6.1 Experimental Setup

Our goal is to compare the utility of these four mechanisms while tuning them all to satisfy ε-d_X-privacy, for the same ε. Observe that Geo-Local achieves pure metric privacy ($\delta = 0$) while the other mechanisms need a $\delta \in (0, 1]$. In the following experiments, we vary the number of users n, showing the improvement of privacy as n increases. It should be noted that our privacy analysis of *RR-Shuffle* (Theorem 3.1) requires a minimum number of users for a specified privacy level ε. Hence, in the following experiments we

Summary of Findings: Utility Loss

Table 2: Synthetic data experiment: MAE

n	ε	Geo Local	RR-Shuffle	Geo-Shuffle	SGDL-Shuffle
50	0.2	1	1.21	0.52	0.1
50	0.5	0.42	0.43	0.05	0.0004
100	0.1	1.2	1.1	0.49	0.03
100	0.2	0.66	0.67	0.18	0.01
100	0.3	0.41	0.41	0.04	0.005
200	0.1	0.92	0.43	0.11	0.007

Table 3: Location data experiment: Euclidean distance (meters)

n	ε	Geo Local	RR-Shuffle	Geo-Shuffle	SGDL-Shuffle
1000	0.15	300	120	40	2
1000	0.2	210	80	40	0.9
1000	0.3	100	40	20	0.04
2000	0.1	190	50	10	1
3000	0.2	130	40	20	0.001
4000	0.3	50	5	1	0.00006

omit the results for RR-Shuffle if the number of users is insufficient for the desired privacy level ε.

6.2 Evaluation on Synthetic Numeric Data

In this experiment, we consider n users who want to privately compute the average of their values. Each user holds a uniformly random integer $\in \{0, \ldots, 1000\}$. We run all three shuffle mechanisms, along with Geo-Local, with $\varepsilon = 0.1$ and $\delta = 10^{-4}$. In this section we use the MAE (Mean Absolute Error) as metric for the utility loss (more utility metrics are available in Section E). Namely if $f(X)$ is the actual result of the query and $M(X)$ is the output of a mechanism M then we define its utility loss as: $U_M = \mathbb{E}[|f(X) - M(X)|]$.

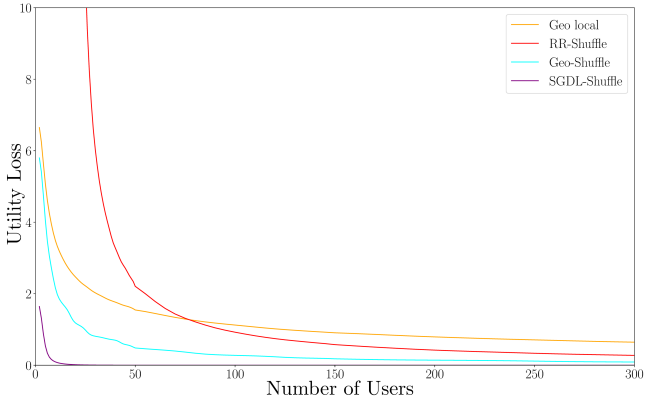


Figure 4: Utility loss with synthetic data. RR-Shuffle cannot achieve $\varepsilon = 0.1$ for $n < 23$

Figure 4 shows the obtained utility level as a function of the number of users n. First, we observe that RR-Shuffle needs at least 23 users for $\varepsilon = 0.1$. Moreover, for $23 \leq n \leq 70$, despite the use of a shuffler, RR-Shuffle actually performs worse than Geo-Local (since the latter is better tailored to metric privacy). However, as the number of users increases, the privacy boost from shuffling becomes larger, and RR-Shuffle outperforms Geo-Local. On the other hand, Geo-Shuffle outperforms both RR-Shuffle and Geo-local for all values of n, offering a substantial improvement, especially for smaller values of n. SGDL-Shuffle offers near-optimal utility even with a small amount of users.

6.3 Evaluation on Real-World Location Data

In this experiment we are going to use GPS locations of addresses in Austin, Texas (Figure 22), from [30], in order to find their *centroid*. To do this, we make a grid of the city of Austin [4], with size of $1000x1000$ squares; each square having a side of 100 meters. Furthermore, each square has a horizontal and a vertical number, from 0 to 1000. Each user i holds a pair of integers x_i, y_i where $x_i, y_i \in \{0, \ldots, 1000\}$ indicates respectively the number of horizontal or vertical square that she belongs. Note that $\{0, 0\}$ is the upper-most left square and $\{1000, 1000\}$ is the bottom-most right square. We average separately all the $x's$ and $y's$ to find the centroid of the data.

We desire to have an $\varepsilon = 0.15$ for all mechanisms and $\delta = 10^{-4}$. If $f(X)$ is the actual centroid and $M(X)$ is the output centroid of a mechanism M, we define the utility loss (in meters) of M as the Euclidean distance, i.e. $U_M = \mathbb{E}[||f(X) - M(X)||_2]$.

Despite running the mechanisms on two different datasets (latitude and longitude), one should observe the correlation between them. We cannot therefore treat these two datasets as independent. Informally, consider that running a mechanism for the horizontal dimension, might leak some information about the vertical dimension. In Section D we show that to achieve (ε, δ) - $||\cdot||_2$ - privacy, we have to apply the mechanism to each dimension with $\frac{\varepsilon}{\sqrt{2}}$ and $\frac{\delta}{2}$. Finally, to tune the value of ε, note that in geo-indistinguishability, the level of privacy is proportional to the desired privacy radius r [1]. Namely each user enjoys $\varepsilon \cdot r$ metric privacy within r. In this experiment we set a privacy radius $r = 600$ meters (or equivalently 6 squares). Thus, in conclusion, we have to run each mechanism for each dimension with $\varepsilon_{\text{run}} = \frac{\varepsilon}{6 \cdot \sqrt{2}} \approx 0.017$ to achieve the target $\varepsilon = 0.15$. Note that for that ε_{run}, *RR-Shuffle* needs at least 776 users, by Theorem 3.1.

Figure 5 shows the resulting utility using box plots, as a function of the number of users. Similar observations as in the previous experiment can be made. The shuffle mechanisms offer better utility than Geo-Local. Between RR-Shuffle and Geo-Shuffle, the latter is utility-wise superior especially when running with fewer users. Moreover, the box plot enables us to see that Geo-Shuffle has less variance compared to RR-Shuffle and Geo-Local. SGDL-Shuffle provides, in this setting, near-optimal utility, as its reported centroid is only a few meters away from the actual one. Finally, based on the observation that Austin is a densely populated area, we also examine in Figure 29 a variation of the experiment, discarding users

[4]Namely, we create a rectangle between 4 points with the following coordinates: (30.091, -98.27048), (30.95534, -98.27048), (30.091, -97.3819) and (30.95534, -97.3819).

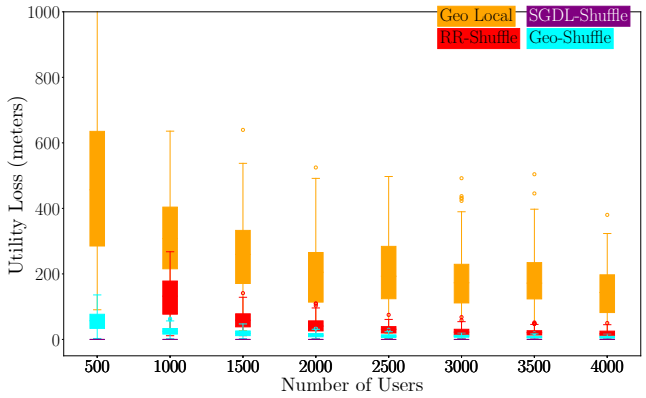


Figure 5: Utility loss with location data (box plot) with $\varepsilon = 0.15$

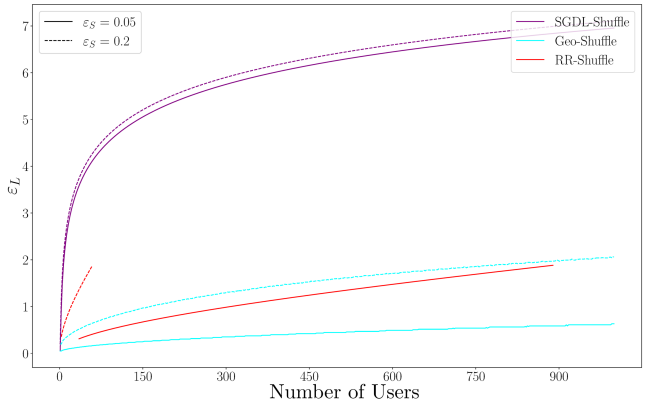


Figure 6: Privacy in the local model (ε_L) when the same level of privacy in the shuffle model (ε_S) is fixed.

living in downtown Austin and keeping only those in the suburbs, which is placed in the Appendix as it displays similar results.

7 Compromised Shuffler

Despite the fact that SGDL-Shuffle provides better utility than the other two proposed mechanisms, surprisingly, it offers virtually no privacy if the shuffler is compromised. In this section, we challenge the assumption that the shuffler is trusted, expanding the adversary model to align more closely with real-world applications, as explained in Section 1. After we formally describe how the privacy of each mechanism is affected, we perform a comparison between them. Proofs can be found in the Appendix.

RR-Shuffle. The privacy guarantees of RR-Shuffle, in the local model, will be:

Theorem 7.1. *RR-Shuffle(δ, p, k, n) is (ε_L, δ) - d_x - private in the local model, if $\lambda_L = p \cdot k$ and $k \geq \lambda_L \geq 14log\frac{4}{\delta}$, where*

$$\varepsilon_L = \sqrt{\frac{32 \log \frac{4}{\delta}}{\lambda_L - \sqrt{2\lambda_L \log \frac{2}{\delta}}}} \tag{7}$$

Geo-Shuffle. It has been shown that the truncated Geo-Local(ε_{geo}) mechanism, which is used in Geo-Shuffle as a Local Randomizer, yields the same privacy guarantees, in the local model, as the non-truncated mechanism [26]. Hence we can trivially show that:

Proposition 5. *Geo-Shuffle($\varepsilon_{\text{geo}}, \delta, n$) is ($\varepsilon_L$,0) - d_x - private in the local model, with $\varepsilon_L = \varepsilon_{\text{geo}}$.*

SGDL-Shuffle. Recall that the amount of noise is decreased with the number of users, since each user samples noise from SGDL $(\frac{1}{n}, e^{-\varepsilon_S})$ in order to have ε_S privacy in the shuffle model, leading to a significantly reduced privacy in the local model.

Theorem 7.2. *SGDL-Shuffle(ε_S, δ, n), is (ε_L,δ) - d_x - private, in the local model, with:*

$$\varepsilon_L = \max_{d_x} \frac{\ln(g(0, d_x(x, x'), e^{-\varepsilon_S}, 1/n))}{d_x(x, x')} \tag{8}$$

for the function g as defined in Theorem 4.1.

Privacy comparison. We fix the same level of privacy in the shuffle model (ε_S) and measure the privacy of each mechanism in the local model (ε_L) if the shuffler is compromised.

For RR-Shuffle and SGDL-Shuffle, ε_L can be directly calculated from the above formulas. On the other hand, for Geo-Shuffle in order to find ε_L we need to find a proper ε_{geo} which if given as a parameter to Geo-Shuffle it will result in the target ε_S privacy (Theorem 4.1). Then, we can use Proposition 5 and set $\varepsilon_L = \varepsilon_{geo}$. Observe that ε_{geo} depends on the number of users; as more users join the system the target ε_S can be reached with a bigger ε_{geo}, due to the increased privacy amplification by shuffling. Figure 6 shows that when the shuffler gets compromised, the privacy of SGDL-Shuffle diminishes significantly. For instance, with 150 users the local privacy ε_L is about 5, while for Geo-Shuffle is about 1. In general, Geo-Shuffle maintains a reasonable level of privacy, surpassing that of RR-Shuffle.

To comprehend the significant difference, recall that the noise applied by each user is sampled from SGDL$(1, e^{-\varepsilon_L})$ in *Geo-Shuffle* and from SGDL$(1/n, e^{-\varepsilon_S})$ in *SGDL-Shuffle*. Therefore, fixing the same ε_S implies that Geo-Shuffle's noise scales with the second parameter of the SGDL distribution, whereas SGDL-Shuffle scales with the first, which depends on the number of users. This obstacle cannot be avoided, as setting the first parameter of SGDL to $1/n$ is necessary in order to take advantage of the property of infinitely divisibility of the geometric mechanism.

A note about the missing values of RR-Shuffle in Figure 6. Recall that RR-Shuffle requires at least a minimum number of users for a specified ε_S (Section 3.2). On the other hand, the number of random bits λ does not depend on n. Hence as the number of users increases, the parameter p (proportion of random bits) decreases, making harder to meet the requirement that $p \cdot k \geq 14log\frac{4}{\delta}$, for the local model. Thus it is only possible to calculate ε_L for a specific range of users. In this experiment we considered $\delta = 0.01$ and $k = 1000$ (recall that k is the maximum value of the secrets domain). Note that δ and k only affect in this comparison, privacy-wise, RR-Shuffle.

8 Primitive Shuffler and Communication Cost

In the proposed mechanisms all users send the exact same number of bits to the shuffler: k bits in RR-Shuffle and $k + 2c$ bits in Geo-Shuffle and SGDL-Shuffle. In other words, each reported message has the same length as the one with the maximum possible value. In this section we discuss this conscious design choice which was based under the assumptions that a) standard shuffling techniques should be applied (Section 2.2) and b) the adversary may monitor the network between the users and the shuffler (Section 2.4).

In the shuffle model literature there is no clear agreement on the exact capabilities of the shuffler. In the original proposal [7], all users send an equal number of bits to the shuffler. Similarly, in [11], a probabilistic rounding step is used to assure same-length messages. On the other hand, in other works the reported number of bits per user varies [3, 20]. In such works, an adversary is assumed to be unable to observe how many bits each user sends, since he only observe the outcome of the shuffler. In other words, the communication between the shuffler and the users is assumed to be entirely hidden from the adversary.

In practice, however, an adversary can often monitor the traffic between the users and the shuffler. Although the actual values remain encrypted, the size of encrypted traffic can still be observed. For example, in the case of MixNets, an adversary with control over the network (such as an ISP or a law enforcement agency) can trivially observe the size of the encrypted traffic, and hence the number of messages sent from each user. Note that this is especially important in metric privacy where the reported (noisy) value is with high probability close to the initial secret. As a consequence, variable-length shuffling which does not leak any information (under such an adversary) is a non-standard operation that cannot be easily achieved via all traditional shuffling implementations.

Moreover, note that if a shuffler is allowed to perform more complicated operations than standard naive shuffling, then there exist various ways of reducing the communication cost. For instance, if the shuffler can also sum the received values (say by substituting the shuffler with a MPC protocol for summation [32]) then there is no reason to employ an inefficient unary encoding. Values can be encoded in binary format, leading to a much smaller communication cost of $\log k$ or resp. $\log(k + 2c)$.

Nonetheless, if one wishes to reduce the communication cost while accepting a privacy loss, they can change the quantization granularity of Algorithm 1. Currently, the bit "1" represents the integer "1", meaning that the integer x is represented by x ones. The granularity Q can change so that each bit of the encoded vector actually represents a larger integer. In Section G we show a possible implementation of this approach. Unfortunately this approach may diminish metric privacy as a high granularity could make the users that hold a big value more easily distinguishable.

9 Conclusion and Future Work

The shuffle model provides a solution to the utility and trust trade-off between the local model and the central model, essentially providing almost the best from both worlds. For this reason, many mechanisms have been proposed in the literature for the shuffle model and standard DP. On the other hand, its implementation in metric privacy had not been studied before.

First we presented RR-Shuffle, inspired by an existing implementation of RR for standard DP. Then, we continued with the geometric mechanism, which is already tailored to metric privacy, and proposed Geo-Shuffle. More specifically, we formalized the way that Geo-Shuffle creates noise, and proved that it follows the Symmetric Generalized Discrete Laplace distribution. Finally, we presented SGDL-Shuffle, which offers optimal error for the summation query.

Our experiments indicate that the shuffle mechanisms provide a utility level that almost matches that of the central model, when executed with many users. When the number of users is small, Geo-Shuffle and SGDL-Shuffle provide sufficiently better utility than Geo-Local.

If the adversary model is expanded, incorporating the corruption of the shuffler by the adversary, then Geo-Shuffle provides significantly better privacy than SGDL-Shuffle. After considering our analysis, one might opt for Geo-Shuffle and tolerate a slight decrease in utility to ensure a level of privacy even if the shuffler gets compromised by the adversary. In scenarios where there is no such a threat, SGDL-Shuffle can be chosen to provide optimal utility.

Our work provided a first study on the shuffle model for metric privacy. As there are many mechanisms that have been studied in the literature for the shuffle model and standard differential privacy, research should continue with them. Additionally, in this work we focused only on *integer summation*. In the future we aim to extend our work with the problem of privately computing histograms. Lastly, changing the granularity Q and studying its impact on metric privacy will offer a formal trade-off between privacy and communication cost.

Acknowledgments

The work of Andreas Athanasiou was supported by the project CRYPTECS, funded by the ANR (project number ANR-20-CYAL-0006) and by the BMBF (project number 16KIS1439). The work of Catuscia Palamidessi was supported by the project HYPATIA, funded by the ERC (grant agreement number 835294).

Finally, we would like to thank Alexis Miller for his precious aid in proving Theorem B.2.

References

[1] Miguel E. André s, Nicolás E. Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi. 2013. Geo-indistinguishability. In *Proceedings of 2013 ACM SIGSAC*. ACM Press. https://doi.org/10.1145/2508859.2516735

[2] Ugur Ilker Atmaca, Sayan Biswas, Carsten Maple, and Catuscia Palamidessi. 2024. A Privacy-Preserving Querying Mechanism with High Utility for Electric Vehicles. *IEEE Open Journal of Vehicular Technology* 5 (Jan. 2024), 262–277. https://doi.org/10.1109/OJVT.2024.3360302

[3] Victor Balcer, Albert Cheu, Matthew Joseph, and Jieming Mao. 2020. Connecting Robust Shuffle Privacy and Pan-Privacy. In *ACM-SIAM Symposium on Discrete Algorithms*.

[4] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. 2019. The Privacy Blanket of the Shuffle Model. In *Advances in Cryptology – CRYPTO 2019*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer International Publishing, Cham, 638–667.

[5] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. 2020. Private Summation in the Multi-Message Shuffle Model. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security* (Virtual Event, USA) *(CCS '20)*. Association for Computing Machinery, New York, NY, USA, 657–676. https://doi.org/10.1145/3372297.3417242

[6] Amos Beimel, Kobbi Nissim, and Eran Omri. 2008. Distributed Private Data Analysis: Simultaneously Solving How and What. In *Advances in Cryptology –*

CRYPTO 2008, David Wagner (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 451–468.
[7] Andrea Bittau, Úlfar Erlingsson, Petros Maniatis, Ilya Mironov, Ananth Raghunathan, David Lie, Mitch Rudominer, Ushasree Kode, Julien Tinnés, and Bernhard Seefeld. 2017. Prochlo: Strong Privacy for Analytics in the Crowd. *CoRR* abs/1710.00901 (2017). arXiv:1710.00901 http://arxiv.org/abs/1710.00901
[8] T-H. Hubert Chan, Elaine Shi, and Dawn Song. 2012. Optimal Lower Bound for Differentially Private Multi-party Aggregation. In *Algorithms – ESA 2012*, Leah Epstein and Paolo Ferragina (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 277–288.
[9] Konstantinos Chatzikokolakis, Miguel E. Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. 2013. Broadening the Scope of Differential Privacy Using Metrics. In *Privacy Enhancing Technologies*, Emiliano De Cristofaro and Matthew Wright (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 82–102.
[10] Albert Cheu. 2021. Differential Privacy in the Shuffle Model: A Survey of Separations. arXiv:2107.11839 https://arxiv.org/abs/2107.11839
[11] Albert Cheu, Adam Smith, Jonathan Ullman, David Zeber, and Maxim Zhilyaev. 2019. Distributed Differential Privacy via Shuffling. In *Advances in Cryptology – EUROCRYPT 2019*. Springer International Publishing, 375–403. https://doi.org/10.1007/978-3-030-17653-2_13
[12] Apple Differential Privacy Team. 2017. Learning with Privacy at Scale. In *Apple Machine Learning Journal*, Vol. 1. Apple.
[13] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting Telemetry Data Privately. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 3574–3583.
[14] Roger Dingledine, Nick Mathewson, and Paul Syverson. 2004. Tor: The Second-Generation Onion Router. In *In Proceedings of the 13th Usenix Security Symposium.*
[15] Devdatt P. Dubhashi and Alessandro Panconesi. 2009. *Concentration of Measure for the Analysis of Randomized Algorithms.* Cambridge University Press. https://doi.org/10.1017/CBO9780511581274
[16] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. *Theory of Cryptography* Vol. 3876, 265–284. https://doi.org/10.1007/11681878_14
[17] Cynthia Dwork and Aaron Roth. 2014. The Algorithmic Foundations of Differential Privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407. https://doi.org/10.1561/0400000042
[18] Úlfar Erlingsson, Vitaly Feldman, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Abhradeep Thakurta. 2019. Amplification by shuffling: from local to central differential privacy via anonymity. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms* (San Diego, California) *(SODA '19)*. Society for Industrial and Applied Mathematics, USA, 2468–2479.
[19] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA) *(CCS '14)*. Association for Computing Machinery, New York, NY, USA, 1054–1067. https://doi.org/10.1145/2660267.2660348
[20] Badih Ghazi, Noah Golowich, Ravi Kumar, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. 2020. Pure Differentially Private Summation from Anonymous Messages. In *1st Conference on Information-Theoretic Cryptography, ITC 2020, June 17-19, 2020, Boston, MA, USA (LIPIcs, Vol. 163)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 15:1–15:23. https://doi.org/10.4230/LIPIcs.ITC.2020.15
[21] Badih Ghazi, Ravi Kumar, Pasin Manurangsi, and Rasmus Pagh. 2020. Private counting from anonymous messages: near-optimal accuracy with vanishing communication overhead. In *Proceedings of the 37th International Conference on Machine Learning (ICML'20)*. JMLR.org, Article 328, 10 pages.
[22] Badih Ghazi, Pasin Manurangsi, Rasmus Pagh, and Ameya Velingker. 2020. Private Aggregation from Fewer Anonymous Messages. In *Advances in Cryptology – EUROCRYPT 2020*, Anne Canteaut and Yuval Ishai (Eds.). Springer International Publishing, Cham, 798–827.
[23] Arpita Ghosh, Tim Roughgarden, and Mukund Sundararajan. 2009. Universally utility-maximizing privacy mechanisms. In *Proceedings of the Forty-First Annual ACM Symposium on Theory of Computing* (Bethesda, MD, USA) *(STOC '09)*. Association for Computing Machinery, New York, NY, USA, 351–360. https://doi.org/10.1145/1536414.1536464
[24] Dov Gordon, Jonathan Katz, Mingyu Liang, and Jiayu Xu. 2022. Spreading the Privacy Blanket: Differentially Oblivious Shuffling for Differential Privacy. In *Applied Cryptography and Network Security: 20th International Conference, ACNS 2022, Rome, Italy, June 20–23, 2022, Proceedings* (Rome, Italy). Springer-Verlag, Berlin, Heidelberg, 501–520. https://doi.org/10.1007/978-3-031-09234-3_25
[25] Slawomir Goryczka and Li Xiong. 2017. A Comprehensive Comparison of Multiparty Secure Additions with Differential Privacy. *IEEE Transactions on Dependable and Secure Computing* 14, 5 (2017), 463–477. https://doi.org/10.1109/TDSC.2015.2484326
[26] Lefki Kacem and Catuscia Palamidessi. 2018. Geometric Noise for Locally Private Counting Queries. In *Proceedings of the 13th Workshop on Programming Languages and Analysis for Security* (Toronto, Canada) *(PLAS '18)*. Association for Computing Machinery, New York, NY, USA, 13–16. https://doi.org/10.1145/3264820.3264827
[27] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What Can We Learn Privately? *SIAM J. Comput.* 40, 3 (2011), 793–826. https://doi.org/10.1137/090756090 arXiv:https://doi.org/10.1137/090756090
[28] Seetha V. Lekshmi and Simi Sebastian. 2014. A Skewed Generalized Discrete Laplace Distribution. *International Journal of Mathematics and Statistics Invention (IJMSI)* 2, 3 (March 2014), 95–102.
[29] Aaron Roth. 2010. *New algorithms for preserving differential privacy.* Ph. D. Dissertation. USA. Advisor(s) Blum, Avrim. AAI3421732.
[30] Donovan Venuto. 2016. *Austin Address with GPS Coords.* Retrieved April 21, 2023 from https://data.world/tronovan/austin-address-with-gps-coords
[31] Stanley L. Warner. 1965. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *J. Amer. Statist. Assoc.* 60, 309 (1965), 63–69. http://www.jstor.org/stable/2283137
[32] Yu Wei, Jingyu Jia, Yuduo Wu, Changhui Hu, Changyu Dong, Zheli Liu, Xiaofeng Chen, Yun Peng, and Shaowei Wang. 2024. Distributed Differential Privacy via Shuffling Versus Aggregation: A Curious Study. *Trans. Info. For. Sec.* 19 (Jan. 2024), 2501–2516. https://doi.org/10.1109/TIFS.2024.3351474
[33] Shuheng Zhou, Katrina Ligett, and Larry Wasserman. 2009. Differential privacy with compression. In *2009 IEEE International Symposium on Information Theory*. 2718–2722. https://doi.org/10.1109/ISIT.2009.5205863

A Metric Privacy of the RR-Shuffle

Theorem A.1. *For any* $\delta > 0, 0 \le p \le 1$ *and* $n \cdot k \ge \lambda \ge 14 log \frac{4}{\delta}$, *let* $\lambda = p \cdot n \cdot k$.
RR-Shuffle(δ, p, k, n) *is* (ε, δ) *-* d_X *- private where*

$$\varepsilon = \sqrt{\frac{32 \log \frac{4}{\delta}}{\lambda - \sqrt{2\lambda \log \frac{2}{\delta}}}}$$

We will use the same approach as the authors of [11] for standard DP, but slightly modify it in order to prove metric privacy.

Observe that the output of the mechanism, as seen by the analyst, includes only a shuffled vector of unary bits. Releasing a shuffled unary vector is privacy-wise equivalent to releasing its sum, since both reveal the same level of information (Proposition 1). It will be more convenient to analyze the algorithm C_λ. The privacy of C_λ carries over to the RR-Shuffle mechanism, because both have the same output space [11].

Algorithm 5: $C_\lambda(x_1, x_2, \ldots, x_n)$ [11]

Input : $(x_1, \ldots, x_n) \in \{0,1\}^n$ parameter $\lambda \in (0, n)$.
Output: $y \in \{0, 1, 2, \ldots, n\}$
Sample $\mathbf{s} \leftarrow Bin(n, \frac{\lambda}{n})$
Define $H_s = \{H \subseteq [n] : |H| = s\}$ and choose $\mathbf{H} \leftarrow H_s$ uniformly at random.
Return: $\mathbf{y} \leftarrow \sum_{i \notin H} x_i + Bin(s, \frac{1}{2})$

We will analyse the privacy of C_λ in three steps. Firstly (Step 1) we will examine a simpler form of the algorithm, C_H where the set of users that respond randomly, H, is given as input. We will prove that for any sufficiently large H, C_H is metric private. Secondly (Step 2), we will prove the algorithm C_s where any sufficiently large value s and $|H|$ chosen randomly with $|H| = s$ keeps the initial privacy of Step 1. In the last step (Step 3) , we will prove that the initial algorithm C_λ is metric private.

Step 1:
Let us begin by finding the privacy of C_H:

Claim A.1. *For any* $\delta > 0$ *and any* $H \subseteq [n]$ *such that* $|H| > 8 log \frac{4}{\delta}$ C_H *is* $(\varepsilon, \frac{1}{2}\delta)$*-* d_X*- private for*

Algorithm 6: $C_H(x_1, x_2, \ldots, x_n)$ [11]

Input : $(x_1, \ldots, x_n) \in \{0,1\}^n$ parameter $H \subseteq [n]$.
Output: $y_H \in \{0, 1, 2, \ldots, n\}$
Let $\mathbf{B} \leftarrow Bin(|H|, \frac{1}{2})$
Return: $\mathbf{y}_H \leftarrow \sum_{i \notin H} x_i + \mathbf{B}$

$$\varepsilon = d_X \cdot \ln\left(1 + \sqrt{\frac{32log\frac{4}{\delta}}{|H|}}\right) < d_X \cdot \left(\sqrt{\frac{32log\frac{4}{\delta}}{|H|}}\right)$$

Proof. We prove this claim using a similar approach as in [11], adapting it to metric privacy. Fix two datasets X , $X' \in \{0,1\}^n$ that have exactly d_X different bits, any $H \subseteq [n]$ such that $|H| > 8log\frac{4}{\delta}$, and any $\delta > 0$. If the points at which X, X' differ all lie within H then of course the two distributions $C_H(X), C_H(X')$ are identical. Hence, the biggest difference between $C_H(X), C_H(X')$ will be when all differing points lie outside of H. W.l.o.g. let us assume that $x_j = 0$ and $x'_j = 1$ for every differing point $j_i \notin H$ where $i \in \{0, 1, \ldots d_X - 1\}$.
We also define $u := \sqrt{\frac{1}{2}|H|log\frac{4}{\delta}}$ and $I_u := \left(\frac{1}{2}|H| - u, \frac{1}{2}|H| + u\right)$, hence using Hoeffding's inequality (Theorem *F.2*): $\mathbb{P}[\mathbf{B} \notin I_u] < \frac{1}{2}\delta$.
We have, for any $W \subseteq \{0, 1, \ldots, n\}$:

$$\begin{aligned}\mathbb{P}[C_H(X) \in W] &= \mathbb{P}[C_H(X) \in W \cap \mathbf{B} \in I_u] + \\ &\quad \mathbb{P}[C_H(X) \in W \cap \mathbf{B} \notin I_u] \\ &\leq \mathbb{P}[C_H(X) \in W \cap \mathbf{B} \in I_u] + \frac{1}{2}\delta \\ &= \sum_{r \in W \cap I_u} \mathbb{P}[\mathbf{B} + \sum_{i \notin H} x_i = r] + \frac{1}{2}\delta \qquad (9)\end{aligned}$$

In order to conclude we have to show that for any H and $r \in W \cap I_u$:

$$\frac{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x_i = r\right]}{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x'_i = r\right]} \leq \left(1 + \sqrt{\frac{32log\frac{4}{\delta}}{|H|}}\right)^{d_X} \qquad (10)$$

Because $x_j = 0$ and $x'_j = 1$ for every differing bit j then $\sum_{i \notin H} x_i = \sum_{i \notin H} x'_i - d_X$. Hence:

$$\begin{aligned}\frac{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x_i = r\right]}{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x'_i = r\right]} &= \frac{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x'_i - d_X = r\right]}{\mathbb{P}\left[\mathbf{B} + \sum_{i \notin H} x'_i = r\right]} \\ &= \frac{\mathbb{P}\left[\mathbf{B} = \left(r - \sum_{i \notin H} x'_i\right) + d_X\right]}{\mathbb{P}\left[\mathbf{B} = \left(r - \sum_{i \notin H} x'_i\right)\right]} \qquad (11)\end{aligned}$$

Let $k = r - \sum_{i \notin H} x'_i$ so that:

$$(11) = \frac{\mathbb{P}[\mathbf{B} = k + d_X]}{\mathbb{P}[\mathbf{B} = k]} \qquad (12)$$

Thus we need to calculate the ratio of the output of the two Binomial distributions. As stated in algorithm C_H, both binomials run with $|H|$ tries and probability to success $\frac{1}{2}$. Hence:

$$\begin{aligned}\frac{\mathbb{P}[\mathbf{B} = k + d_X]}{\mathbb{P}[\mathbf{B} = k]} &= \frac{\binom{|H|}{k+d_X}\frac{1}{2}^{k+d}\frac{1}{2}^{|H|-k-d_X}}{\binom{|H|}{k}\frac{1}{2}^{k}\frac{1}{2}^{|H|-k}} \\ &= \frac{\binom{|H|}{k+d_X}\frac{1}{2}^{|H|}}{\binom{|H|}{k}\frac{1}{2}^{|H|}} \\ &= \frac{\binom{|H|}{k+d_X}}{\binom{|H|}{k}} \\ &= \frac{\frac{|H|!}{(k+d_X)!(|H|-k-d_X)!}}{\frac{|H|!}{k!(|H|-k)!}} \\ &= \frac{k!(|H|-k)!}{(k+d_X)!(|H|-k-d_X)!} \\ &= \frac{k!(|H|-k-d_X)!\prod_{i=0}^{d_X-1}(|H|-k-i)}{(k)!\prod_{i=1}^{d_X}(k+i)(|H|-k-d_X)!} \\ &= \frac{\prod_{i=0}^{d_X-1}(|H|-k-i)}{\prod_{i=1}^{d_X}(k+i)} \\ &= \prod_{i=0}^{d_X-1}\frac{|H|-k-i}{k+i+1} \qquad (13)\end{aligned}$$

The fraction of the product maximizes when i = 0. By Claim 4.4 of [11] it is proven that:

$$\frac{|H|-k}{k+1} \leq 1 + \sqrt{\frac{32log\frac{4}{\delta}}{|H|}} \qquad (14)$$

Since we have d_X products the bound will be:

$$(13) \leq \left(1 + \sqrt{\frac{32log\frac{4}{\delta}}{|H|}}\right)^{d_X} \qquad (15)$$

Note that, $\ln\left(1 + \sqrt{\frac{32log\frac{4}{\delta}}{|H|}}\right)^{d_X} = d_X \cdot \ln\left(1 + \sqrt{\frac{32log\frac{4}{\delta}}{|H|}}\right)$. However, we need to prove that $d_X \cdot \ln\left(1 + \sqrt{\frac{32log\frac{4}{\delta}}{|H|}}\right) < d_X \cdot \left(\sqrt{\frac{32log\frac{4}{\delta}}{|H|}}\right)$.
For notational convenience, let $y = \sqrt{\frac{32log\frac{4}{\delta}}{|H|}}$.

$$\begin{aligned}d_X \cdot ln(1+y) &< d_X \cdot y \\ ln(1+y) &< y \\ 1 + y &< e^y\end{aligned}$$

which is true for $y > 0$ and completes the proof. □

Step 2:
Now we consider the case that H is a random subset of $[n]$ but with a fixed size s. We will analyse the following algorithm:

Claim A.2. *For any $\delta > 0$ and any $8log\frac{4}{\delta} < s$ C_s is $(\varepsilon, \frac{1}{2}\delta)$ - d_X - private for*

$$\varepsilon = d_X \cdot \left(\sqrt{\frac{32\log\frac{4}{\delta}}{s}}\right)$$

Algorithm 7: $C_s(x_1, x_2, \ldots, x_n)$ [11]

Input : $(x_1, \ldots, x_n) \in \{0,1\}^n$ *parameters* $\in \{0, 1, 2, \ldots, n\}$
Output: $y_s \in \{0, 1, 2, \ldots, n\}$
Define $H_s = \{H \subseteq [n] : |H| = s\}$ and choose $\mathbf{H} \leftarrow H_s$ uniformly at random.
Return: $\mathbf{y}_s \leftarrow C_H(x)$

Proof. We follow the corresponding proof from [11]. Fix two datasets, $X \sim X' \in \{0,1\}^n$ which differ by d_X bits. Let H be the realization of $\mathbf{H}$. For shorthand, let us name the privacy that we proved that $C_H(X)$ has as $\varepsilon_0(|H|) = d_X \cdot \left(\sqrt{\frac{32log\frac{4}{\delta}}{|H|}}\right)$. The algorithm $C_s(X)$ selects H uniformly random from H_s and then runs $C_H(X)$. Because $s \geq 8log\frac{4}{\delta}$, for any s, **Claim** *A*.1 is directly applicable. Hence, for any such s:

$$\mathbb{P}[C_s(X) \in W] = e^{\varepsilon_0(s)}\mathbb{P}[C_s(X') \in W] + \tfrac{1}{2}\delta \leftrightarrow$$
$$\mathbb{P}[C_s(X) \in W] = exp(d_X \cdot \left(\sqrt{\tfrac{32\log\frac{4}{\delta}}{s}}\right))\mathbb{P}[C_s(X') \in W] + \tfrac{1}{2}\delta$$

□

Note that, in contrast to [11], we cannot achieve a privacy boost for C_s. In standard differential privacy the randomness of H improves the privacy parameters. However, in metric privacy, since d can be arbitrary large, the boosting factor cannot be guaranteed to be achieved.

Step 3:
Now we are ready to prove the privacy of C_λ. We prove that with high probability, s is almost as large as λ, using the same technique as in [11]. Let us restate the claim:

Claim A.3. *For any $\delta > 0$ and $\lambda \geq 14log\frac{4}{\delta}$, C_λ is (ε, δ) - d_X - private where*

$$\varepsilon = \left(d_X \cdot \sqrt{\frac{32\log\frac{4}{\delta}}{\lambda - \sqrt{2\lambda\log\frac{2}{\delta}}}}\right)$$

Proof. Fix two datasets X, X' $in\{0,1\}^n$ that differ by d bits and any $W \subseteq [n]$.

$$\mathbb{P}[C_\lambda(X) \in W] = \mathbb{P}[C_\lambda(X) \in W \cap \mathbf{s} \geq \lambda - \sqrt{2\lambda log\frac{2}{\delta}}] +$$
$$\mathbb{P}[C_\lambda(X) \in W \cap \mathbf{s} < \lambda - \sqrt{2\lambda log\frac{2}{\delta}}]$$
$$\leq \mathbb{P}[C_\lambda(X) \in W \cap \mathbf{s} \geq \lambda - \sqrt{2\lambda log\frac{2}{\delta}}] + \frac{1}{2}\delta \textbf{ [Chernoff Bound]}$$
$$= \sum_{s \geq \lambda - \sqrt{2\lambda log\frac{2}{\delta}}} \mathbb{P}[C_s(X) \in W] \cdot \mathbb{P}\mathbf{s} = s] + \frac{1}{2}\delta \tag{16}$$

As λ is sufficiently large, $\lambda - \sqrt{\lambda log\frac{2}{\delta}} > 8log\frac{4}{\delta}$. Hence, we can apply **Claim** *A*.2 to each term of the sum.

$$\mathbb{P}[C_s(X) \in W] \leq d_X \cdot \left(\sqrt{\tfrac{32log\frac{4}{\delta}}{s}}\right) \cdot \mathbb{P}[C_s(X') \in W] + \tfrac{1}{2}\delta$$

For shorthand, let $\varepsilon_1(s) := d_X \cdot \left(\sqrt{\frac{32log\frac{4}{\delta}}{s}}\right)$. Thus:

$$(16) \leq \left(\sum_{s \geq \lambda - \sqrt{2\lambda log\frac{2}{\delta}}} \left[e^{\varepsilon_1(s)}\mathbb{P}[C_s(X') \in W] + \frac{1}{2}\delta\right] \cdot \mathbb{P}[\mathbf{s} = s]\right) + \frac{1}{2}\delta$$
$$\leq \left(\sum_{s \geq \lambda - \sqrt{2\lambda log\frac{2}{\delta}}} e^{\varepsilon_1(s)}\mathbb{P}[C_s(X') \in W] \cdot \mathbb{P}[\mathbf{s} = s]\right) + \delta$$
$$\max_{s \geq \lambda - \sqrt{2\lambda log\frac{2}{\delta}}} e^{\varepsilon_1(s)} \cdot \mathbb{P}[C_\lambda(X') \in W] + \delta \tag{17}$$

Observe that $\varepsilon_1(s)$ decreases with s. Thus the above is maximized at the lower bound of s:

$$\mathbb{P}[C_\lambda(X) \in W] \leq \exp\left(d_X \cdot \sqrt{\frac{32\log\frac{4}{\delta}}{\lambda - \sqrt{2\lambda\log\frac{2}{\delta}}}}\right) \cdot \mathbb{P}[C_\lambda(X') \in W] + \delta$$

□

A.1 Utility of RR-Shuffle

Observe that n users holding k bits each is privacy-wise equivalent to $n \cdot k$ users holding each only 1 bit. That is because the shuffler shuffles all bits together, hence every bit can end up in every position with the same probability.

We will hence treat the problem as having $n \cdot k$ users, each with 1 bit. We again follow the method of [11].

Throughout the rest of this section, for notational convenience, let $P_{n\cdot k,\lambda}(X)$ for $X \in \{0,1\}^n$ denote the RR-Shuffle protocol, which consists of three mechanisms:

- $\mathbf{R}_{n\cdot k,\lambda}(x)$ **(Randomiser)** which denotes the output of the RR, run on some bit $x \in \{0,1\}$ with parameter $p = \frac{\lambda}{n\cdot k}$.
- ***S* (Shuffler)** which shuffles the $n \cdot k$ bits reported respectively by $n \cdot k$ users.
- ***A* (Analyst)** which sees the output of the Shuffler.

Firstly let us make two claims that will serve as building blocks for the proof:

Claim A.4. *[11] For any $n \in \mathbb{N}$, $0 < \lambda \leq n \cdot k$:*

$$\mathbb{E}[R_{n\cdot k,\lambda}(x)] = \frac{\lambda}{2 \cdot n \cdot k} + \left(1 - \frac{\lambda}{n \cdot k}\right) \cdot x \tag{18}$$
$$Var[R_{n\cdot k,\lambda}(x)] = \frac{\lambda}{2 \cdot n \cdot k} \cdot \left(1 - \frac{\lambda}{n \cdot k}\right) \tag{19}$$

Proof. The proof is reported from [11], with slight changes to reflect metric privacy. Observe that if $x = 0$ then the user can report ***1*** only if he responds randomly. Hence the probability to report 1 follows the Bernoulli Random variable with probability $\frac{1}{2}\ \frac{\lambda}{n\cdot k}$. The variance of this random variable will be $\frac{\lambda}{2\cdot n\cdot k} \cdot \left(1 - \frac{\lambda}{n\cdot k}\right)$. A symmetric argument can be made to report ***0*** when $x = 1$. Thus:

$$\mathbb{E}[R_{n\cdot k,\lambda}(x)] =$$
$$\frac{\lambda}{n \cdot k} \cdot \mathbb{E}\left[Ber\left(\frac{1}{2}\right)\right] + \left(1 - \frac{\lambda}{n \cdot k}\right) \cdot x =$$
$$\frac{\lambda}{2 \cdot n \cdot k} + \left(1 - \frac{\lambda}{n \cdot k}\right) \cdot x \tag{20}$$

Using this claim, we can compute the expected value and variance of $P_{n\cdot k,\lambda}$ because the output of the protocol is the sum of all the reported messages.

Claim A.5. *[11] For any $n \in \mathbb{N}$, $0 < \lambda \leq n \cdot k$:*

$$\mathbb{E}[P_{n\cdot k,\lambda}(X)] = \sum_{i=1}^{n\cdot k} R_{n\cdot k,\lambda}(x_i) \tag{21}$$

$$Var[P_{n\cdot k,\lambda}(X)] = \big(\frac{n\cdot k}{n\cdot k-\lambda}\big)^2 \cdot \frac{\lambda}{2}\big(1-\frac{\lambda}{2\cdot n\cdot k}\big) \tag{22}$$

□

Theorem A.2. *Let P be RR-Shuffle(δ, p, k, n) and $\lambda = p\cdot n\cdot k$. For every $n \in \mathbb{N}$, $0 < \beta \leq 1$, $nk > \lambda \geq 2log\frac{2}{\beta}$ and $X = \{0,1\}^{nk}$:*

$$\mathbb{P}\left[\left|P(X) - \textstyle\sum_i x_i\right| > \frac{nk}{nk-\lambda}\sqrt{2\lambda \log\frac{2}{\beta}}\right] < \beta$$

Proof. To prove this theorem we use the same technique as the authors of [11]. Let $X \in \{0,1\}^n$ and d_i denote $R_{n\cdot k,\lambda}(x_i) - \frac{\lambda}{2\cdot n\cdot k} - \big(1-\frac{\lambda}{n\cdot k}\big)\cdot x_i$. Observe that $-1 < d_i < 1$. From the previous claim, we can conclude that $\mathbb{E}[d_i] = 0$ and $\mathrm{Var}[d_i] = \frac{\lambda}{2\cdot n\cdot k}\big(1-\frac{\lambda}{2\cdot\lambda\cdot k}\big)$. Let $1 \geq \beta > 0$. Note that $\mathrm{Var}[d_i] > \frac{4}{9\cdot n\cdot k}\log\frac{2}{\beta}$, because λ is sufficiently large. Thus, we can use Bernstein's inequaility (**Theorem** F.3):

$$\mathbb{P}\left[|\sum_{i=1}^{n\cdot k} d_i| > \sqrt{2\lambda\big(1-\frac{\lambda}{2nk}\big)\log\frac{2}{\beta}}\right] < \beta \tag{23}$$

Moreover, if $y_i = R_{n\cdot k,\lambda}(x_i)$ and $f(x) = \sum_i x_i$:

$$\sum_i d_i = \sum_i \big(y_i - \frac{\lambda}{2nk} - (1-\frac{\lambda}{nk})\cdot x_i\big) = \big(\sum_i y_i\big) - \frac{\lambda}{2} - \big(1-\frac{\lambda}{nk}\big)\cdot f(x))$$

$$\frac{nk}{nk-\lambda}\sum_i d_i = \frac{nk}{nk-\lambda}\left(\Big(\sum_i y_i\Big) - \frac{\lambda}{2}\right) - f(x) = P_{n\cdot k,\lambda}(X) - f(x) \tag{24}$$

Hence (23) will became, after substitution:

$$\mathbb{P}\left[|P_{n\cdot k,\lambda}(X) - f(x)| > \frac{nk}{nk-\lambda}\sqrt{2\lambda\big(1-\frac{\lambda}{2nk}\big)\log\frac{2}{\beta}}\right] < \beta \tag{25}$$

Since $\lambda > 0$, we can conclude. □

A.2 RR-Shuffle in the local model

Let us now study the case when the shuffler is compromised:

Theorem A.3. *RR-Shuffle(δ, p, k, n) is (ε_L, δ) - d_x - private in the local model, if $\lambda_L = p\cdot k$ and $k \geq \lambda_L \geq 14log\frac{4}{\delta}$, where*

$$\varepsilon_L = \sqrt{\frac{32\log\frac{4}{\delta}}{\lambda_L - \sqrt{2\lambda_L\log\frac{2}{\delta}}}} \tag{26}$$

Proof. Recall that summing unary bits is privacy-wise equivalent to shuffling them (Proposition 1). If we assume that some user i with value x_i has to report, according to the protocol, a unary vector b_i of size k then the adversary will make the same observations about x_i if he sees directly all k bits of b_i or if k users report each 1 bit of b_i to the shuffler and the shuffled output is released to him. Hence it is a special case of Theorem 3.1 for $n = k$ users, each sending 1 bit. Setting $\lambda_L = p\cdot k$ reflects this case. □

B Metric Privacy of the Geo-Shuffle mechanism

First let us prove that shuffling a binary vector is privacy-wise equivalent to revealing its sum.

Proposition 1. *Let a mechanism $K(X) = \sum_{x\in X} R(x)$ be (ε, δ) - d_X - private for a dataset X and R be a Local Randomizer. Then, the shuffle model mechanism with a shuffler S (as defined in Section 2.2) which uses the unary encoding $\mathcal{U}(x, r)$ (Algorithm 1):*

$$M(X) = S\Big(\{\mathcal{U}(R(x), r) : x \in X\}\Big)$$

is also (ε, δ) - d_X - private.

Proof. First, let us estimate the probability that a shuffler outputs a particular bit vector. Let v be the output of a shuffler $S(y_1, \ldots y_m)$ with sum (number of 1's) $\hat{v}$. Observe that v is only described by $\hat{v}$ and by its arrangement of bits. Hence if A is the event to have $\hat{v}$ 1's and B is the event to achieve a given sequence of bits with $\hat{v}$ 1's, then:

$$\mathbb{P}[S(y_1, \ldots y_m) = v] = \mathbb{P}(A \cap B) = \mathbb{P}(A)\cdot\mathbb{P}(B)$$

Let us proceed to our case and consider two datasets X and X' of size n with distance $d_X(X, X')$. Here v will be the output of $S(\{\mathcal{U}(R(x), r) : x \in X\})$. Recall that each $\mathcal{U}(R(x), r)$ returns a bit vector of size r and the shuffler S shuffles all the bits together creating a single bit vector v. Hence v will have a length of $n\cdot r$ with $\hat{v} = \sum_{x\in X} R(x)$.

Therefore, $\mathbb{P}(A) = \mathbb{P}\big[\sum_{x\in X}(R(x)) = \hat{v}\big]$ and $\mathbb{P}(B) = \frac{1}{\binom{nr}{\hat{v}}}$ for every given sequence of bits with $\hat{v}$ ones, since shuffling randomly permutates the vector.

Hence for every vector v with length $n\cdot r$ and sum $\hat{v}$:

$$\mathbb{P}[S(y_1, \ldots y_m) = v] = \mathbb{P}(A)\cdot\mathbb{P}(B) = \frac{\mathbb{P}\big[\sum_{x\in X}(R(x)) = \hat{v}\big]}{\binom{nr}{\hat{v}}} \tag{27}$$

So far we have worked only for a single vector v with $\hat{v}$ ones. Since the definition of metric privacy is over sets (i.e. any subset of the output space of the mechanism), we should properly expand our scope of vectors. Let us define $\mathcal{V}$ as a set of unary vectors (the output space of the shuffler), and $\hat{\mathcal{V}}$ as a set of integers representing the sum (number of 1's) of the corresponding (i.e. same position) vector in $\mathcal{V}$. For example, if $\mathcal{V} = \{\{0,1,1\},\{0,0,0\},\{1,1,1\}\}$, then $\hat{\mathcal{V}} = \{2,0,3\}$. The relationship between $\mathcal{V}$ and $\hat{\mathcal{V}}$ is the same as between v and $\hat{v}$: given $\mathcal{V}$ one can design a deterministic function to produce $\hat{\mathcal{V}}$.

However, if $\hat{\mathcal{V}}$ is given and the goal is to produce a target $\mathcal{V}$, then this function is probabilistic, following the same operation of the shuffler. This function goes as follows, for each $\hat{v} \in \hat{\mathcal{V}}$: a vector v with a known size is created with exactly $\hat{v}$ ones in (uniformly) random positions and it is finally placed in $\mathcal{V}$.

Let us consider this event to get a specific $\mathcal{V}$, given $\hat{\mathcal{V}}$. If we denote this event as C, then by using the previously computed probability on B, for each $\hat{v} \in \hat{\mathcal{V}}$:

$$P(C) = \prod_{\hat{v}\in\hat{\mathcal{V}}} \frac{1}{\binom{nr}{\hat{v}}} \tag{28}$$

Hence we can expand Equation (27) with sets:

$$\mathbb{P}[S(y_1,\dots y_m)\in\mathcal{V}]=\mathbb{P}\Big[\sum_{x\in X}(R(x))\in\hat{\mathcal{V}}\Big]\cdot\prod_{\hat{v}\in\hat{V}}\frac{1}{\binom{nr}{\hat{v}}} \tag{29}$$

Moreover:

$$\begin{aligned}\frac{\mathbb{P}[M(X)\in\mathcal{V}]}{\mathbb{P}[M(X')\in\mathcal{V}]}&=\frac{\mathbb{P}\Big[S(\{\mathcal{U}(R(x),r):x\in X\})\in\mathcal{V}\Big]}{\mathbb{P}\Big[S(\{\mathcal{U}(R(x'),r):x'\in X'\})\in\mathcal{V}\Big]}\\&=\frac{\mathbb{P}\Big[\sum_{x\in X}(R(x))\in\hat{\mathcal{V}}\Big]\cdot\prod_{\hat{v}\in\hat{V}}\frac{1}{\binom{nr}{\hat{v}}}}{\mathbb{P}\Big[\sum_{x'\in X'}(R(x'))\in\hat{\mathcal{V}}\Big]\cdot\prod_{\hat{v}\in\hat{V}}\frac{1}{\binom{nr}{\hat{v}}}}\\&=\frac{\mathbb{P}[K(X)\in\hat{\mathcal{V}}]}{\mathbb{P}[K(X')\in\hat{\mathcal{V}}]}\end{aligned} \tag{30}$$

Because K is (ε,δ) - d_X - private:

$$\begin{aligned}&\mathbb{P}[K(X)\in\hat{\mathcal{V}}]\le e^{\varepsilon\cdot d_X(X,X')}\mathbb{P}[K(X')\in\hat{\mathcal{V}}]+\delta\Leftrightarrow\\&\frac{\mathbb{P}[K(X)\in\hat{\mathcal{V}}]}{\mathbb{P}[K(X')\in\hat{\mathcal{V}}]}\le e^{\varepsilon\cdot d_X(X,X')}+\frac{\delta}{\mathbb{P}[K(X')\in\hat{\mathcal{V}}]}\end{aligned} \tag{31}$$

Furthermore, note that:

$$\begin{aligned}\mathbb{P}[M(X')\in\mathcal{V}]&=\mathbb{P}\Big[\sum_{x'\in X'}(R(x'))\in\hat{\mathcal{V}}\Big]\cdot\prod_{\hat{v}\in\hat{V}}\frac{1}{\binom{nr}{\hat{v}}}\\&\le\mathbb{P}\Big[\sum_{x'\in X'}(R(x'))\in\hat{\mathcal{V}}\Big]\\&=\mathbb{P}[K(X')\in\hat{\mathcal{V}}]\end{aligned} \tag{32}$$

Finally, by combining Equation (30) and Equation (31) we can prove that M is also (ε,δ) - d_X - private. For every subset $\mathcal{V}$ of the output space of M:

$$\begin{aligned}\mathbb{P}[M(X)\in\mathcal{V}]&\le e^{\varepsilon\cdot d_X(X,X')}\mathbb{P}[M(X')\in\mathcal{V}]+\frac{\delta\cdot\mathbb{P}[M(X')\in\mathcal{V}]}{\mathbb{P}[K(X')\in\hat{\mathcal{V}}]}\\&\le e^{\varepsilon\cdot d_X(X,X')}\mathbb{P}[M(X')\in\mathcal{V}]+\delta\text{ [using (32)]}\end{aligned}$$

Note that [11] has also showed a similar result; it suffices to study the privacy of the sum of the bits when considering a mechanism in the shuffle model with a Local Randomizer which outputs one bit. □

Now we can turn our attention in bounding the probability of the geometric mechanism to report a notably large/small value (formally expressed by c) by $\frac{\delta}{2}$. Recall that our goal is to bound the probability that $x_i+N_i+c\notin[0,k+2c]$ but since we want to bound that probability regardless of x_i, we can reduce the problem by bounding $N_i\in[-c,c]$. One might simply view this as taking the worst possible case each time i.e. $x_i=0$ or $x_i=k$ for the lower and upper bound respectively.

Let us restate our theorem:

Proposition 2. *Let $N_i\sim G(0,\varepsilon_{\text{geo}})$, $\delta\in(0,1]$ and :*

$$c=\left\lceil-\frac{\ln\left[\frac{\left(1+e^{-\varepsilon_{geo}}\right)\left(1-\sqrt[n]{1-\frac{\delta}{2}}\right)}{2}\right]}{\varepsilon_{geo}}\right\rceil$$

Then:

$$1-\prod_{i=1}^{n}\mathbb{P}\big[N_i\in[-c,c]\big]\le\frac{\delta}{2}$$

Proof. The probability that the output of the geometric mechanism is greater then c will be:

$$\begin{aligned}&\sum_{i=c+1}^{\infty}\frac{1-e^{-\varepsilon_{geo}}}{1+e^{-\varepsilon_{geo}}}\cdot e^{-\varepsilon_{geo}i}\le\\&\sum_{i=c}^{\infty}\frac{1-e^{-\varepsilon_{geo}}}{1+e^{-\varepsilon_{geo}}}\cdot e^{-\varepsilon_{geo}i}=\\&\frac{1-e^{-\varepsilon_{geo}}}{1+e^{-\varepsilon_{geo}}}\sum_{i=0}^{\infty}e^{-\varepsilon_{geo}(i+c)}=\\&\frac{1-e^{-\varepsilon_{geo}}}{1+e^{-\varepsilon_{geo}}}\cdot\frac{e^{-\varepsilon_{geo}\cdot c}}{1-e^{-\varepsilon_{geo}}}=\quad\text{[Geometric Series]}\\&\frac{e^{-\varepsilon_{geo}\cdot c}}{1+e^{-\varepsilon_{geo}}}\end{aligned} \tag{33}$$

Because the geometric mechanism is symmetric, the probability that its output is less then $-c$ will also be equal to (33). Hence, the probability that the output is outside of the interval $[-c,c]$ will be:

$$2\cdot\frac{e^{-\varepsilon_{geo}\cdot c}}{1+e^{-\varepsilon_{geo}}} \tag{34}$$

It needs to be assured that the probability of all n users getting an output of the geometric mechanism outside of $[-c,c]$ is at most $\frac{\delta}{2}$. Hence:

$$\begin{aligned}&1-\left(1-2\cdot\frac{e^{-\varepsilon_{geo}\cdot c}}{1+e^{-\varepsilon_{geo}}}\right)^n\le\frac{\delta}{2}\leftrightarrow\\&2\cdot\frac{e^{-\varepsilon_{geo}\cdot c}}{1+e^{-\varepsilon_{geo}}}\le1-\sqrt[n]{1-\frac{\delta}{2}}\leftrightarrow\\&c\ge-\frac{\ln\left[\frac{\left(1+e^{-\varepsilon_{geo}}\right)\left(1-\sqrt[n]{1-\frac{\delta}{2}}\right)}{2}\right]}{\varepsilon_{geo}}\end{aligned} \tag{35}$$

Finally, we need to use the ceiling function because, in our case, c represents a number of bits. □

The fact that we examined the tail bound of only the noise means that we did not take into account at all the input dataset which leads us to the next corollary.

Corollary 1. *If n users run Geo-Shuffle($\varepsilon_{\text{geo}},\delta,n$), each with a secret x_i, the probability that at least one of them reports a truncated x_i' is at most $\delta/2$, regardless of each x_i.*

Using the following Proposition 3 it suffices to study the privacy of the case where nobody truncates.

Proposition 3. *Let M be a mechanism and H be an event independent of the input of M such that $\Pr[H]\ge1-\delta_1$. Moreover, assume that when H occurs M is (ϵ,δ_2)- d_X - private, namely for all X,X':*

$$\Pr[M(X)\in S|H]\le e^{\epsilon\cdot d_X(X,X')}\Pr[M(X')\in S|H]+\delta_2$$

If K is a mechanism that behaves like M when H occurs then K is $(\epsilon,\delta_1+\delta_2)$ - d_X - private.

Proof. If O is the output space of K then for any $S\subseteq O$:

$$\begin{aligned}
&\Pr[K(X) \in S] = \Pr[K(X) \in S|H]\Pr[H] + \Pr[K(X) \in S|\neg H]\Pr[\neg H] \\
&\leq \Pr[K(X) \in S|H]\Pr[H] + \delta_1 \qquad [\Pr[H] \geq 1 - \delta_1] \\
&= \Pr[M(X) \in S|H]\Pr[H] + \delta_1 \qquad [K \text{ behaves as } M \text{ when } H \text{ occurs}] \\
&\leq (e^{\epsilon \cdot d_X(X,X')}\Pr[M(X') \in S|H] + \delta_2)\Pr[H] + \delta_1 [M \text{ is private under } H] \\
&\leq e^{\epsilon \cdot d_X(X,X')}\Pr[K(X') \in S|H]\Pr[H] + \delta_1 + \delta_2 \\
&\leq e^{\epsilon \cdot d_X(X,X')}\left(\Pr[K(X') \in S|H]\Pr[H] + \Pr[K(X') \in S|\neg H]\Pr[\neg H]\right) \\
&+ \delta_1 + \delta_2 \\
&= e^{\epsilon \cdot d_X(X,X')}\Pr[K(X') \in S] + \delta_1 + \delta_2
\end{aligned}$$

□

In the case of Geo-Shuffle (which can be considered as K in the phrasing of Proposition 3), we set H as the event that nobody has to truncate their obfuscated value which is independent of the input dataset (Corollary 1). We set M as the mechanism which corresponds to the special case of Geo-Shuffle where no user truncates their reported value. Thus we set $\delta_1 = \delta/2$ and therefore it suffices to set the remaining δ_2 to $\delta/2$ as well.

Finally note that it does not matter what K does when H does not occur.

We continue by showing the following claim, which will be necessary in the next proof.

Claim B.1. *Let G denote the geometric mechanism. Then the noise added by $G(x, \varepsilon_{geo})$ can be expressed as the difference between two geometric distributions, with parameter $(x, 1 - e^{-\varepsilon_{geo}})$.*

Proof. We will use the law of total probability. Let X, Y be i.i.d. variables that follow the geometric distribution with a parameter p. For any $k \in \mathbb{N}$:

$$\begin{aligned}
\mathbb{P}[X - Y = k] &= \sum_{i=-\infty}^{+\infty} \mathbb{P}[X = i + k] \cdot \mathbb{P}[Y = i] = \\
&\sum_{i=0}^{+\infty} \mathbb{P}[X = i + k] \cdot \mathbb{P}[Y = i] = \qquad [X, Y \geq 0\,] \\
&\sum_{i=0}^{+\infty} p(1-p)^{i+k} \cdot p(1-p)^i = \\
&p^2(1-p)^k \cdot \sum_{i=0}^{+\infty} (1-p)^{2^i} = \\
&p^2(1-p)^k \cdot \frac{1}{1-(1-p)^2} = \\
&(1-p)^k \cdot \frac{p^2}{(1-1+p)(1+1-p)} = \\
&(1-p)^k \cdot \frac{p}{(2-p)} = \\
&\frac{1-e^{-\varepsilon_{geo}}}{1+e^{-\varepsilon_{geo}}} \cdot e^{-\varepsilon_{geo} k} \qquad [\text{let } p = 1 - e^{-\varepsilon_{geo}}]
\end{aligned} \tag{36}$$

The difference between X and Y can also be negative:

$$\begin{aligned}
\mathbb{P}[X - Y = -k] &= \sum_{i=-\infty}^{+\infty} \mathbb{P}[X = k] \cdot \mathbb{P}[Y = i + k] = \\
&\sum_{i=0}^{+\infty} p(1-p)^{i+k} \cdot p(1-p)^i = \\
&\frac{1-e^{-\varepsilon_{geo}}}{1+e^{-\varepsilon_{geo}}} \cdot e^{-\varepsilon_{geo} k} [\text{as in (36)}]
\end{aligned} \tag{37}$$

□

Next, let us prove why the noise added by the mechanism when nobody truncates can be described by the Symmetric Generalized Discrete Laplace distribution [28].

Proposition 6. *Let X be an integer dataset with n users and $\delta \in (0, 1]$. Then:*

$$\mathbb{P}\left[\textit{Geo-Shuffle}(\varepsilon_{\text{geo}}, \delta, n) = \sum_{x \in X}(x + c) + N_{total}\right] \geq 1 - \frac{\delta}{2} \tag{38}$$

where $N_{total} \sim SGDL(n, e^{-\varepsilon_{\text{geo}}})$.

Proof. We again consider the case where nobody truncates their output, which happens w.p. $1 - \frac{\delta}{2}$ (Proposition 2).

Recall that shuffling unary bits is privacy-wise equivalent to revealing their sum (Proposition 1). Let $\mathcal{G}$ denote the geometric distribution and N_{total} denote the noise added by the mechanism. We are going to show that N_{total} follows the Symmetric Generalized Discrete Laplace distribution.

From Claim B.1, N_{total} will be :

$$N_{total} = \sum_{i=1}^{n} \left[\mathcal{G}(x_i, 1 - e^{-\varepsilon_{geo}}) - \mathcal{G}(x_i, 1 - e^{-\varepsilon_{geo}})\right] \tag{39}$$

It is well known that the Negative-Binomial distribution can represent the sum of i.i.d. random variables that follow the geometric distribution:

$$N_{total} = NB(n, 1 - e^{-\varepsilon_{geo}}) - NB(n, 1 - e^{-\varepsilon_{geo}}) \tag{40}$$

The difference of two i.i.d. Negative Binomial random variables with the same parameters, (n, p) can be expressed as the symmetric Generalized Discrete Laplace distribution (SGDL) with parameters $(n, 1 - p)$[28]. Hence:

$$N_{total} = SGDL(n, e^{-\varepsilon_{geo}}) \tag{41}$$

□

We continue by proving the privacy of the Symmetric Generalized Discrete Laplace distribution.

Theorem B.1. *Let $a = -\frac{ln(\frac{\delta}{4} \cdot [\frac{(1-p)^2}{(1-pe^t)(1-pe^{-t})}]^{-\beta})}{t}$ and $t = 2 \cdot \frac{\varepsilon}{\sqrt{n}}$ and a distance d_X. The Symmetric Generalized Discrete Laplace Distribution $(n, e^{-\varepsilon_{geo}})$ is $(\varepsilon, \frac{\delta}{2})$ - d_X - private, with $\varepsilon = \frac{ln(g(-\alpha, d_X))}{d_X}$, where:*

$$g(r, d_X) = max\{h(r, d_X), \tfrac{1}{h(r, d_X)}\}, \text{ and}$$
$$h(r, d_X) = \frac{\sum_{k=|r|}^{\infty} \binom{\beta+k-1}{k}\binom{\beta+k-|r|-1}{k-|r|} p^k p^{k-|r|}}{\sum_{k=|r-d_X|}^{\infty} \binom{\beta+k-1}{k}\binom{\beta+k-|r-d_X|-1}{k-|r-d_X|} p^k p^{k-|r-d_X|}}$$

Proof. Let $Y \sim SGDL(\beta, p)$ with the following Probability Mass Function:

$$\mathbb{P}(Y = m) = (1-p)^{2\cdot\beta} \sum_{k=|m|}^{\infty} \binom{\beta+k-1}{k}\binom{\beta+k-|m|-1}{k-|m|} p^k p^{k-|m|}$$

Let $t = 2 \cdot \frac{\varepsilon_{geo}}{\sqrt{n}}$. Also let:

$P\left[SGDL(1/n, e^{-\varepsilon}) \notin [-c, c]\right] \leq \delta$

$$a = -\frac{ln(\frac{\delta}{4} \cdot [\frac{(1-p)^2}{(1-pe^t)(1-pe^{-t})}]^{-\beta})}{t} \tag{42}$$

The moment generating function of SGDL(p, β) is [28]:

$$M_y(t) = \Big[\frac{(1-p)^2}{(1-pe^t)(1-pe^{-t})}\Big]^\beta, -logp < t < logp$$

Applying the Chernoff Bound yields the inequality:

$$P(Y \geq a) \leq e^{-ta} M_Y(t) = e^{-ta}\Big[\frac{(1-p)^2}{(1-pe^t)(1-pe^{-t})}\Big]^\beta = \frac{\delta}{4} \tag{43}$$

Let $I := [-a, a]$. By Chernoff's Bound, for every $Y \sim SGDL(\beta, p)$:

$$\mathbb{P}(Y \geq a) \leq \tfrac{\delta}{4}$$

Because SGDL is even, also:

$$\mathbb{P}(Y \leq -a) \leq \tfrac{\delta}{4}$$

Hence $\mathbb{P}(Y \notin I) \leq \frac{\delta}{2}$.

Consider now two datasets, X and X' that have a distance of $d_X(X, X')$. It will be more convenient to assume, w.l.o.g. that for some x, the sum of X is $\lfloor x - \frac{d_X}{2} \rfloor$ and the sum of X' is $\lfloor x + \frac{d_X}{2} \rfloor$. Note that we have to use the floor function to make the sum be an integer, since both datasets contain only integer numbers.

$\forall W \subset \mathbb{Z}$ we have:

$$\begin{aligned}\mathbb{P}[SGDL(X) \in W] = \mathbb{P}[SGDL(X) \in W \cap SGDL(X) \in I] + \\ \mathbb{P}[SGDL(X) \in W \cap SGDL(X) \notin I] \leq \\ \mathbb{P}[SGDL(X) \in I] + \frac{\delta}{2}\end{aligned} \tag{44}$$

For every $r \in I \cap W$, let us calculate the following ratio:

$$\frac{\mathbb{P}[\lfloor r - \frac{d_X}{2} \rfloor = SGDL(\beta, p)]}{\mathbb{P}[\lfloor r + \frac{d_X}{2} \rfloor = SGDL(\beta, p)]} = \frac{\sum_{k=|\lfloor r - \frac{d_X}{2} \rfloor|}^{\infty} \binom{\beta+k-1}{k}\binom{\beta+k-|\lfloor r-\frac{d_X}{2}\rfloor|-1}{k-|\lfloor r-\frac{d_X}{2}\rfloor|} p^k p^{k-|\lfloor r-\frac{d_X}{2}\rfloor|}}{\sum_{k=\lfloor |r+\frac{d_X}{2}| \rfloor}^{\infty} \binom{\beta+k-1}{k}\binom{\beta+k-\lfloor |r+\frac{d_X}{2}| \rfloor-1}{k-\lfloor |r+\frac{d_X}{2}| \rfloor} p^k p^{k-\lfloor |r+\frac{d_X}{2}| \rfloor}} \tag{45}$$

We would like to find the proper values of ε that satisfy the following inequality:

$$(45) \leq e^{\varepsilon \cdot d} \tag{46}$$

Note that the inverse ratio should also be calculated, because, depending on d_X, it could be larger. We are looking to find an upper bound, hence we are interested in the maximum of these two ratios. Let us denote

$$h(r, d_X) = (45) \tag{47}$$

and

$$g(r, d_X) = max\{h(r, d_X), \frac{1}{h(r, d_X)}\} \tag{48}$$

Note that the function h also depends on β and p which we omit for notational convenience. For the remainder of this section we will hence write $h(r, d_X)$ and not $h(r, d_X, p, \beta)$. For the same reason we write $g(r, d)$ and not $g(r, d_X, p, \beta)$. We will however retain the form $g(r, d_X, p, \beta)$ in the statement of Theorem 4.1 to help the reader understand the relationship between the variables without referring to the proof.

Then we have the following theorem:

Theorem B.2. *Let $d_X \geq 1, \beta \geq 1$ and $0 < p < 1$. Then, for any fixed d_X, $g(r, d_X)$ as a function of r has the following behavior:*
(1) If d_X is even, $g(r, d_X)$ is monotonically increasing for $r \geq 0$. If d_X is odd, $g(r, d_X)$ is monotonically increasing for $r \geq 1$
(2) If d_X is even, $g(r, d_X)$ is symmetric w.r.t. the axis $x = 0$. If d_X is odd, $g(r, d_X)$ is symmetric w.r.t. the axis $r = 1/2$.
(3) if d_X is even then $g(r, d_X)$ has its minimum in $r = 0$ and $g(0, d) = 1$.
If d_X is odd then $g(0, d_X)$ and $g(1, d_X)$ are the minima.

Proof follows in Appendix B below.

From Theorem B.2 we can immediately conclude that g has a maximum value in $[-a, a]$ of $g(-\alpha, d_X)$.

$$(45) \leq g(-\alpha, d_X) \tag{49}$$

Hence the privacy loss of SGDL will be:

$$\varepsilon = \frac{ln(g(-\alpha, d_X))}{d_X}$$

□

Recall that the probability of each user reporting a value outside of $[-c, c]$ is bounded by $\frac{\delta}{2}$, as proved in Proposition 2. For that reason it sufficed to study the privacy (using ε and the remaining $\frac{\delta}{2}$) in the case where no user has to truncate his output. Since this noise is described by SGDL, the problem reduced to showing that the SGDL distribution with parameters $(n, e^{-\varepsilon_{geo}})$ is $(\varepsilon, \frac{\delta}{2})$ d_X- private. Hence we are now ready to prove the privacy of the Geo-Shuffle Mechanism:

Theorem 4.1. *Let $\delta \in (0, 1]$. The Geo-Shuffle($\varepsilon_{\text{geo}}, \delta, n$) mechanism is (ε, δ)- d_X - private, with:*

$$\varepsilon = \max_{d_X} \frac{ln(g(-\alpha, d_X(X, X'), e^{-\varepsilon_{\text{geo}}}, n))}{d_X(X, X')} \tag{5}$$

where

$$g(r, d, e^{-\varepsilon_{\text{geo}}}, n) = \max\{h(r, d, e^{-\varepsilon_{\text{geo}}}, n), \frac{1}{h(r, d, e^{-\varepsilon_{\text{geo}}}, n)}\}$$

$$h(r, d, p, n) = \frac{\sum_{k=|\lfloor r-\frac{d}{2}\rfloor|}^{\infty} \binom{n+k-1}{k}\binom{n+k-|\lfloor r-\frac{d}{2}\rfloor|-1}{k-|\lfloor r-\frac{d}{2}\rfloor|} p^k p^{k-|\lfloor r-\frac{d}{2}\rfloor|)}}{\sum_{k=|\lfloor r+\frac{d}{2}\rfloor|}^{\infty} \binom{n+k-1}{k}\binom{n+k-|\lfloor r+\frac{d}{2}\rfloor|-1}{k-|\lfloor r+\frac{d}{2}\rfloor|} p^k p^{k-|\lfloor r+\frac{d}{2}\rfloor}}$$

$$t = 2 \cdot \frac{\varepsilon_{\text{geo}}}{\sqrt{n}} \qquad p = e^{-\varepsilon_{\text{geo}}}$$

$$\alpha = -\frac{\ln(\frac{\delta}{4} \cdot [\frac{(1-p)^2}{(1-pe^t)(1-pe^{-t})}]^{-n})}{t}$$

Proof. From Theorems 2 and B.1 we can conclude that the Geo-Shuffle mechanism will be (ε, δ) metric private with:

$$\varepsilon = \frac{ln(g(-\alpha, d_X))}{d_X} \tag{50}$$

After conducting numerous experiments (presented in the end of Appendix B) we see that $d_X = 1$ maximizes the ratio. We can hence conclude that:

$$\varepsilon = \ln(g(-\alpha, 1))$$

□

In the rest of this appendix, we include the missing proof of Theorem B.2. Before proving Theorem B.2 we will prove that:

Theorem B.3. *If $r \geq d_X/2$. $r, d_X \in N$ and $\beta \geq 1$, $0 < p < 1$ and*

$$f(r,d_X) = \frac{\sum_{k=\lfloor r-\frac{d_X}{2}\rfloor}^{\infty} \binom{\beta+k-1}{k}\binom{\beta+k-\lfloor r-\frac{d_X}{2}\rfloor-1}{k-\lfloor r-\frac{d_X}{2}\rfloor} p^k p^{k-\lfloor r-\frac{d_X}{2}\rfloor})}{\sum_{k=\lfloor r+\frac{d_X}{2}\rfloor}^{\infty} \binom{\beta+k-1}{k}\binom{\beta+k-\lfloor r+\frac{d_X}{2}\rfloor-1}{k-\lfloor r+\frac{d_X}{2}\rfloor} p^k p^{k-\lfloor r+\frac{d_X}{2}\rfloor}}$$

then, (1) : $f(r,d_X) \geq 1$ for all r, d_X as above and (2) $f(r,d_X)$ is monotonically increasing in r.

Proof. First of all, we note that $f(r,d_X)$ is well defined, as the summations in the numerator and denominator both converge. This is because $p < 1$, the term p^k eventually vanishes, even when multiplied by the combinatorial factors. We will show that it is sufficient to prove the theorem only for $d_X = 1$. In fact, $f(r,d_X)$ is of the form $\frac{\phi(r-d_X)}{\phi(r)}$, and:

$\frac{\phi(r-d_X)}{\phi(r)} = \frac{\phi(r-d_X)}{\phi(r-(d_X-1))} \cdot \frac{\phi(r-(d_X-1))}{\phi(r-(d_X-2)} \cdots \frac{\phi(r-1)}{\phi(r)}$

Furthermore, if we prove (2) for every $r \geq d_X$ (and $d = 1$) then it is sufficient to prove (1) for $d_X = 1$ and $r = 1$.

We will now prove (1) for $d_X = 1$ and $r = 1$. We start with the following observation:

Remark B.4. *If $a, b \geq 0$ and $\frac{a}{b} \geq \frac{c}{d_X} \geq \frac{b}{a}$ then $\frac{a}{b} \geq 1$*

Proof. Immediate: $\frac{a}{b} \geq \frac{b}{a} \rightarrow a^2 \geq b^2$ since $a, b \geq 0$, we can conclude. □

Now, let us rewrite $f(1,1)$ in a more convenient form:

$$f(1,1) = \frac{\sum_{k=0}^{\infty}\binom{\beta+k-1}{k}\cdot\binom{\beta+k-1}{k}p^{2k}}{\sum_{k=1}^{\infty}\binom{\beta+k-1}{k}\cdot\binom{\beta+k-2}{k-1}p^{2k-1}} =$$

$$\frac{(\sum_{k=1}^{\infty}\binom{\beta+k-1}{\beta-1}\cdot\binom{\beta+k-2}{\beta-1}p^{2k}\cdot\frac{\beta+k-1}{k}) + 1}{\sum_{k=1}^{\infty}\binom{\beta+k-1}{\beta-1}\cdot\binom{\beta+k-2}{\beta-1}p^{2k-1}} \geq$$

$$\frac{\sum_{k=1}^{\infty}\binom{\beta+k-1}{\beta-1}\cdot\binom{\beta+k-2}{\beta-1}p^{2k}\cdot\frac{\beta+k-1}{k}}{\sum_{k=1}^{\infty}\binom{\beta+k-1}{\beta-1}\cdot\binom{\beta+k-2}{\beta-1}p^{2k}} =$$

$$\frac{\sum_{k=0}^{\infty}\binom{\beta+k}{\beta-1}\cdot\binom{\beta+k-1}{\beta-1}p^{2k}\cdot\frac{\beta+k}{k+1}}{\sum_{k=0}^{\infty}\binom{\beta+k}{\beta-1}\cdot\binom{\beta+k-1}{\beta-1}p^{2k}} = \mathbf{A} \tag{51}$$

Analogously we have:

$$\frac{1}{f(1,1)} = \frac{\sum_{k=0}^{\infty}\binom{\beta+k-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}\frac{\beta+k}{1+k}p}{\sum_{k=0}^{\infty}\binom{\beta+k-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}} = \mathbf{B}$$

□

By Remark B.4, it is sufficient to show that $A \geq B$. We note that A and B are both convex sums of the same monotonically non-increasing function of k. Namely:

$$A = \sum_{k=0}^{\infty} c_k\phi(k) \text{ and}$$

$$B = \sum_{k=0}^{\infty} d_k\phi(k)$$

where:

$$c_k = \frac{\binom{\beta+k}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}}{\sum_{k=0}^{\infty}\binom{\beta+k}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}}$$

$$d_k = \frac{\binom{\beta+k-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}}{\sum_{k=0}^{\infty}\binom{\beta+k-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}}$$

the c_k's and d_k's are convex coefficients, in the since that $\forall k\ c_k \geq 0$ and $\sum_{k=0}^{\infty} c_k = 1$

The same holds for all the d_k's.

Furthermore, $\phi(k) = \frac{\beta+k}{1+k}p$ is monotonically non increasing in k. The next of the proof follows from the next two lemmata:

Lemma B.5. *If $\sum_{k=0}^{n} c_k \geq \sum_{k=0}^{n} d_k\ \forall m$, with c_k, d_k being convex coefficients and $\phi(k)$ is monotonically non increasing and $\phi(n)$, $\sum_{k=0}^{n} c_k\phi(k)$ and $\sum_{k=0}^{n} d_k\phi(k)$, converge in n then $\sum_{k=0}^{n} c_k\phi(k) \geq \sum_{k=0}^{n} d_k\phi(k)$*

Proof. Let us recall Abel's transformation (discrete version of integration by parts):

$$\sum_{k=0}^{N} a_k b_k = a_N B_N - \sum_{n=0}^{N-1} B_n(a_{n+1} - a_n), \text{when } B_n = \sum_{k=0}^{n} b_k \tag{52}$$

Let us define $C_n = \sum_{k=0}^{n} c_k$ and $D_n = \sum_{k=0}^{n} d_n$.

By Abel's transformation we have:

$$\sum_{k=0}^{N} c_k\phi(k) = \phi_N C_N + \sum_{n=0}^{N-1} C_n(\phi(n) - \phi(n-1)), \text{and} \tag{53}$$

$$\sum_{k=0}^{N} d_k\phi(k) = \phi_N D_N + \sum_{n=0}^{N-1} D_n(\phi(n) - \phi(n-1)) \tag{54}$$

Hence:

$$\sum_{k=0}^{N} c_k\phi(k) - \sum_{k=0}^{N} d_k\phi(k) =$$

$$\phi(N)C_N - \phi(n)D_N + \sum_{n=0}^{N-1}(C_n - D_n)(\phi(n) - \phi(n+1)) \tag{55}$$

And for $n \rightarrow \infty$:

$$\sum_{k=0}^{\infty} c_k\phi(k) - \sum_{k=0}^{\infty} c_k\phi(k) =$$

$$\phi(\infty)c_\infty - \phi(\infty)D_\infty + \sum_{n=0}^{\infty}(C_n - D_n)(\phi(n) - \phi(n+1)) =$$

$$\phi(\infty) - \phi(\infty) + \sum_{n=0}^{\infty}(C_n - D_n)(\phi(n) - \phi(n+1)) \geq$$

$$0 \text{ because } C_n - D_n \geq 0 \text{ and } \phi(n) - \phi(n+1) \geq 0 \tag{56}$$

Lemma B.6. *If $\phi(k)$ is monotonically non-increasing function, and $\sum_{k=0}^{n} a_k$, $\sum_{k=0}^{n} a_k\phi(k)$ converge in n, then for all $n \geq 0$:*

$$\frac{\sum_{k=0}^{n} a_k\phi(k)}{\sum_{k=0}^{\infty} a_k\phi(k)} \geq \frac{\sum_{k=0}^{n} a_k}{\sum_{k=0}^{\infty} a_k} \tag{57}$$

Proof. It is sufficient to show that, for all n,m with $m \geq n$, we have:

$$\frac{\sum_{k=0}^{n} a_k\phi(k)}{\sum_{k=0}^{m} a_k\phi(k)} \geq \frac{\sum_{k=0}^{n} a_k}{\sum_{k=0}^{m} a_k} \tag{58}$$

To prove the latter, observe that the denominators are positive and:

$$\sum_{k=0}^{n} a_k\phi(k)\sum_{i=0}^{m} a_i = \sum_{k=0}^{n} a_k\phi(k)\sum_{i=0}^{n} a_i + \sum_{k=0}^{n} a_k\phi(k)\sum_{i=n+1}^{m} a_i \tag{59}$$

$$\sum_{k=0}^{n} a_k \sum_{i=0}^{m} a_i\phi(i) = \sum_{k=0}^{n} a_k \sum_{i=0}^{n} a_i\phi(i) + \sum_{k=0}^{n} a_k \sum_{i=n+1}^{m} a_i\phi(i) \quad (60)$$

□

Furthermore:

$$\sum_{k=0}^{n} a_k\phi(k) \sum_{i=0}^{n} a_i = \sum_{k=0}^{n} a_k \sum_{i=0}^{n} a_i\phi(i) \quad (61)$$

while:

$$\sum_{k=0}^{n} a_k\phi(k) \sum_{i=n+1}^{m} a_i \geq \sum_{k=0}^{m} a_k \sum_{i=n+1}^{m} a_i\phi(i) \quad (62)$$

because $\phi(k) \geq \phi(i), \forall k \in \{0, \ldots, n\} \forall i \in \{n+1, \ldots, m\}$

Let $a_k = \binom{\beta+k-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}$ and $\phi(k) = \frac{\beta+k}{1+k}p$

. We have $c_k = \frac{a_k\phi(k)}{\sum_{h=0}^{\infty} a_h\phi(h)}$ and $d_k = \frac{a_k}{\sum_{h=0}^{\infty} a_h}$ and $c_k, d_k, \phi(k)$ and a_k satisfy the conditions of lemmata 1 and 2, hence we can conclude the proof of (1), since we shown that $A \geq B$ □

We will now prove (2), namely that $f(r, 1)$ is monotonically increasing in r.

Proof. Let us start by rewriting $f(r, 1)$ in a more convenient form:

$$f(r,1) = \frac{\sum_{k=r-1}^{\infty}\binom{\beta+k-1}{\beta-1}\binom{\beta+k-(r-1)-1}{\beta-1}p^{2k-(r-1)}}{\sum_{k=r}^{\infty}\binom{\beta+k-1}{\beta-1}\binom{\beta+k-r-1}{\beta-1}p^{2k-r}} = \frac{\sum_{k=1}^{\infty}\binom{\beta+k+r-2}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k+1} + \binom{\beta+r-2}{\beta-1}p}{\sum_{k=0}^{\infty}\binom{\beta+k+r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}} = \frac{\sum_{k=0}^{\infty}\binom{\beta+k-r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k+3}\frac{\beta+k}{1+k} + \binom{\beta+r-2}{\beta-1}p}{\sum_{k=0}^{\infty}\binom{\beta+k+r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}} \quad (63)$$

It is sufficient to show that:
(a) the function

$$g(r) = \frac{\sum_{k=0}^{\infty}\binom{\beta+k+r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}\frac{\beta+k}{1+k}}{\sum_{k=0}^{\infty}\binom{\beta+k+r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}} \quad (64)$$

is not decreasing in r. and:
(b) the function:

$$h(r) = \frac{\binom{\beta+r-2}{\beta-1}}{\sum_{k=0}^{\infty}\binom{\beta+k+r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}} \quad (65)$$

is not decreasing in r.

Let us start with (a). We will show that $g(r+1) \geq g(r)$. We note that:

$$g(r+1) = \frac{\sum_{k=0}^{\infty}\binom{\beta+k+r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}\frac{\beta+k+r}{1+k+r}\frac{\beta+k}{1+k}}{\sum_{k=0}^{\infty}\binom{\beta+k+r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}\frac{\beta+k+r}{1+k+r}} \quad (66)$$

By applying Lemma 1 and Lemma 2, we can conclude. □

We will now show (b).

Proof. To do this, it is equivalent to show that $p(r) \geq \mu(r)$ when:

$$p(r) = \frac{\binom{\beta+r-1}{\beta-1}}{\binom{\beta+r-2}{\beta-1}} \text{ and } \mu(r) = \frac{\sum_{k=0}^{\infty}\binom{\beta+k+r}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}}{\sum_{k=0}^{\infty}\binom{\beta+k+r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}} \quad (67)$$

We have that: $p(r) = \frac{\beta+r-1}{r}$
and:

$$\mu(r) = \frac{\sum_{k=0}^{\infty}\binom{\beta+k+r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}\frac{\beta+k+r}{1+k+r}}{\sum_{k=0}^{\infty}\binom{\beta+k+r-1}{\beta-1}\binom{\beta+k-1}{\beta-1}p^{2k}} \quad (68)$$

Hence $\mu(r)$ has the form of $\sum_{k=0}^{\infty} c_k\frac{\beta+k+r}{1+k+r}$, when the $c_k's$ are convex coefficients. Since $\frac{\beta+k+r}{1+k+r} \leq \frac{\beta+r-1}{r}$ This concludes the proof. □

Now let us prove Theorem B.2 by expanding Theorem B.3. Let us consider the following ratio:

$$h(r,d_X) = \frac{\sum_{k=\lfloor r-\frac{d_X}{2}\rfloor}^{\infty}\binom{\beta+k-1}{k}\binom{\beta+k-\lfloor r-\frac{d_X}{2}\rfloor-1}{k-\lfloor r-\frac{d_X}{2}\rfloor}p^k p^{k-\lfloor r-\frac{d_X}{2}\rfloor)}}{\sum_{k=\lfloor r+\frac{d_X}{2}\rfloor}^{\infty}\binom{\beta+k-1}{k}\binom{\beta+k-\lfloor r+\frac{d_X}{2}\rfloor-1}{k-\lfloor r+\frac{d_X}{2}\rfloor}p^k p^{k-\lfloor r+\frac{d_X}{2}\rfloor}} \quad (69)$$

and the function:

$$g(r,d_X) = max\{h(r,d_X), \frac{1}{h(r,d_X)}\} \quad (70)$$

We want to study the behavior of $g(r, d_X)$ for all values of r. Let us restate the theorem:

Theorem B.2. *Let $d_X \geq 1, \beta \geq 1$ and $0 < p < 1$. Then, for any fixed d_X, $g(r, d_X)$ as a function of r has the following behavior:*
(1) If d_X is even, $g(r, d_X)$ is monotonically increasing for $r \geq 0$. If d_X is odd, $g(r, d_X)$ is monotonically increasing for $r \geq 1$
(2) If d_X is even, $g(r, d_X)$ is symmetric w.r.t. the axis $x = 0$. If d_X is odd, $g(r, d_X)$ is symmetric w.r.t. the axis $r = 1/2$.
(3) if d_X is even then $g(r, d_X)$ has its minimum in $r = 0$ and $g(0, d) = 1$. If d_X is odd then $g(0, d_X)$ and $g(1, d_X)$ are the minima.

Proof. Before we present the proof, it is worth showing a plot in Figure 7 that illustrates the behavior of $g(r, d_X)$ depending on whether d_X is even or odd:

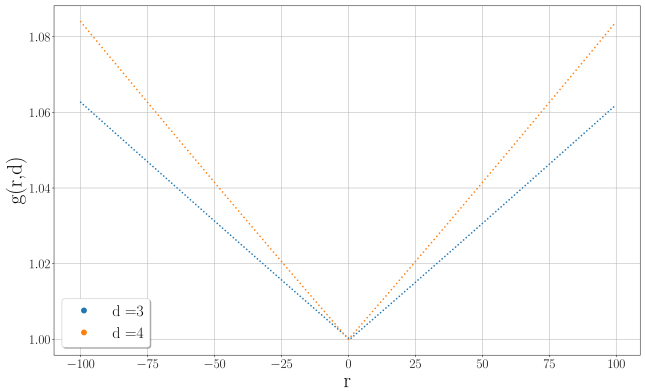


Figure 7: Behavior of $g(r, d_X)$ for an even d and an odd d with $\beta = 100$, $p = 0.81$

To show how the symmetry changes depending on whether d_X is even or d_X is odd, we present in Figure 8 the values of $g(r, d_X)$ only when

$-5 \leq r \leq 5$. Observe that for even values of d_X the axis of symmetry is $r = 0$ but for an odd d_X the axis of symmetry is at $r = 0$ or $r = 1$.

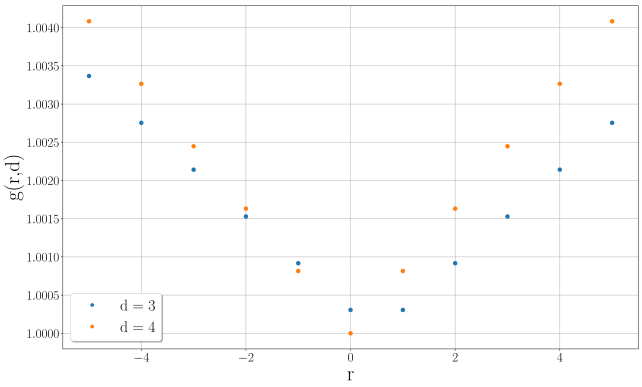


Figure 8: Symmetry of $g(r, d_X)$ with $\beta = 100$, $p = 0.81$

For notation convenience let:

$$C1(\beta,k) = \binom{\beta+k-1}{k}$$
$$C2(\beta,k,r) = \binom{\beta+k-r-1}{k-r}$$
$$P1(k) = p^k$$
$$P2(k,r) = p^{k-r} \tag{71}$$

First we prove (1). Let us begin with the case where d_X is even. We consider the following two cases:

a) $r \geq d_X/2$. In this case $h(r, d_X) = f(r, d_X)$ and (1) follows from Theorem B.3

b) $0 \leq r < d_X/2$: We have:

$$h(r,d_X) = \frac{\sum_{k=d_X/2-r} C1(\beta,k)\cdot C2(\beta,k,d_X/2-r),\cdot P1(k)\cdot P2(k,d_X/2-r)}{\sum_{k=r+d_X/2} C1(\beta,k)\cdot C2(\beta,k,r+d_X/2),\cdot P1(k)\cdot P2(k,r+d_X/2)} = f(d_X/2, 2r). \tag{72}$$

Theorem B.3 can be directly applied because $d_X/2 \geq \frac{2r}{2}$

Now let us examine the case where d is odd:

a) $r \geq d_X/2$. Again in this case (1) follows from Theorem B.3

b) $1/2 \leq r < d_X/2$:

$$h(r,d_X) = \frac{\sum_{k=\frac{d_X+1}{2}-r} C1(\beta,k)\cdot C2(\beta,k,\frac{d_X+1}{2}-r),\cdot P1(k)\cdot P2(k,\frac{d_X+1}{2}-r)}{\sum_{k=r+\frac{d_X-1}{2}} C1(\beta,k)\cdot C2(\beta,k,r+\frac{d_X-1}{2}),\cdot P1(k)\cdot P2(k,r+\frac{d_X-1}{2})} = f(d_X/2, 2r-1). \tag{73}$$

Theorem B.3 can be applied because $d_X/2 \geq \frac{2r-1}{2}$ since $r \leq \frac{d_X+1}{2}$

We will now prove (2). Let us begin with the case where d is even. Being symmetric w.r.t. $x = 0$ means that, if $\frac{r+r'}{2} = 0$, which means that if $r = -r'$, then $g(r, d_X) = g(r', d_X)$

a)For $r \leq -d_X/2$. In this case:

$$h(r,d_X) = \frac{\sum_{k=d_X/2-r} C1(\beta,k)\cdot C2(\beta,k,d_X/2-r),\cdot P1(k)\cdot P2(k,d_X/2-r)}{\sum_{k=-r-d_X/2} C1(\beta,k)\cdot C2(\beta,k,-r-d_X/2),\cdot P1(k)\cdot P2(k,-r-d_X/2)} = \frac{\sum_{k=d_X/2+r'} C1(\beta,k)\cdot C2(\beta,k,d_X/2+r'),\cdot P1(k)\cdot P2(k,d_X/2+r')}{\sum_{k=r'-d_X/2} C1(\beta,k)\cdot C2(\beta,k,r'-d_X/2),\cdot P1(k)\cdot P2(k,r'-d_X/2)} = \frac{1}{h(r',d_X)} = \frac{1}{f(r',d_X)} \leq 1$$

from part (1), case a, of the proof and Theorem B.3 (74)

b)For $-d_X/2 \leq r \leq 0$. In this case:

$$h(r,d_X) = \frac{\sum_{k=d_X/2-r} C1(\beta,k)\cdot C2(\beta,k,d_X/2-r),\cdot P1(k)\cdot P2(k,d_X/2-r)}{\sum_{k=r+d_X/2} C1(\beta,k)\cdot C2(\beta,k,r+d_X/2),\cdot P1(k)\cdot P2(k,r+d_X/2)} = \frac{\sum_{k=r'+d_X/2} C1(\beta,k)\cdot C2(\beta,k,r'+d_X/2),\cdot P1(k)\cdot P2(k,r'+d_X/2)}{\sum_{k=d_X/2-r'} C1(\beta,k)\cdot C2(\beta,k,d_X/2-r'),\cdot P1(k)\cdot P2(k,d_X/2-r')} = \frac{1}{f(d_X/2,2r')} \leq 1$$ from part (1), case b, of the proof and Theorem B.3 (75)

Now for odd values of d, the symmetry is at the axis $x = 1/2$, hence $\frac{r+r'}{2} = \frac{1}{2}$, thus $r = 1 - r'$.

Remark B.7. *If $d \in N$ is odd and $r + r' = 1$ where $r, r' \in Z$, then:*

$$||\lfloor r - d/2\rfloor| = |\lfloor r' + d/2\rfloor|$$
$$|\lfloor r + d/2\rfloor| = |\lfloor r' - d/2\rfloor|$$

Proof. Let us formally define the outputs of the floor function:

$\lfloor r - d/2\rfloor = m$ where m is defined as: $max_i(m_i \in Z | m_i <$ [5] $r - d/2)$
$\lfloor r + d/2\rfloor = n$ where n is defined as: $max_i(n_i \in Z | n_i < r + d/2)$

Also let:

$\lfloor r' - d/2\rfloor = m'$ where m' is defined as: $max_i(m'_i \in Z | m'_i < r' - d/2)$
$\lfloor r' + d/2\rfloor = n'$ where n' is defined as: $max_i(n'_i \in Z | n'_i < r' + d/2)$

We will show that $|n'| = |m|$:
Since $r + r' = 1$ and $r, r' \in Z$, one of r, r' must be positive and the other must be negative, or zero. First, let us examine the later.
W.l.o.g. assume that $r = 0$ and $r' = 1$:

$$|n'| = n' = \lfloor 1 + d/2\rfloor$$
and:
$$|m| = -m = -\lfloor -d/2\rfloor = |n'|$$

Similarly we can prove that $|n'| = |m|$
Now, w.l.o.g assume that r is negative and r' is positive.

$$|n'| = n' = \lfloor r' + d/2\rfloor = max_i(n'_i \in Z | n'_i < r' + d/2) = max_i(n'_i \in Z | n'_i < 1 + d/2 - r)$$

Also:

$$|m| = -m = -\lfloor r - d/2\rfloor = -max_i(m'_i \in Z | -m'_i < r - d/2)$$

Hence m will be equal to the previous integer before $r - d/2$ and thus $|m|$ will be equal to the next integer after $d/2 - r$. On the other hand, $|n'|$ will be equal to the previous integer before $1 + d/2 - r$. Since $(d/2 - r) \notin Z$ between $d/2 - r$ and $1 + d/2 - r$ there is exactly one integer which is equal to $|m|$ and $|n'|$.
Similarly we can prove that $|m'| = |n|$. □

Based on this remark, if $r < 1/2$:

$$h(r,d) = \frac{\sum_{k=|m|} C1(\beta,k)\cdot C2(\beta,k,|m|),\cdot P1(k)\cdot P2(k,|m|)}{\sum_{k=|n|} C1(\beta,k)\cdot C2(\beta,k,|n|),\cdot P1(k)\cdot P2(k,|n|)} = \frac{\sum_{k=|n'|} C1(\beta,k)\cdot C2(\beta,k,|n'|),\cdot P1(k)\cdot P2(k,|n'|)}{\sum_{k=|m'|} C1(\beta,k)\cdot C2(\beta,k,|m'|),\cdot P1(k)\cdot P2(k,|m'|)} = \frac{1}{h(r',d)} \leq 1$$ from part (1) of the proof (76)

[5] In the actual definition of the floor function, the inequality is not strict. But since $(r - d/2) \notin Z$, and $(r + d/2) \notin Z$ we can use a strict inequality in our definitions.

Hence we have proven that for any value of d: $g(r,d) = h(r',d) = g(r',d)$.

We now prove (3).
When d is even, g(r,d) is symmetric w.r.t. $r = 0$ from (a) and (b) we have that $g(r,d)$ is monotonically decreasing for $r < 0$ and monotonically increasing for $r > 0$. If d is odd, the axis of symmetry is $r = 1/2$, so the points of minimum are the integers immediately before and immediately after $1/2$, namely $r = 0$ and $r = 1$.

□

Finally, we will experimentally show that Equation (50) maximizes when $d_X = 1$.

Let $K(r,d) = \frac{ln(g(r,d))}{d}$. Note that $K(r,d)$ is a discrete function and depends on g, hence its output again depends on whether d is even or odd. Let us begin by showing in Figure 9 what happens at the maximum possible value of $r = -1$, assuming $n = 100$ users running the *Geo-Shuffle* mechanism with $\varepsilon_{geo} = 0.2$.

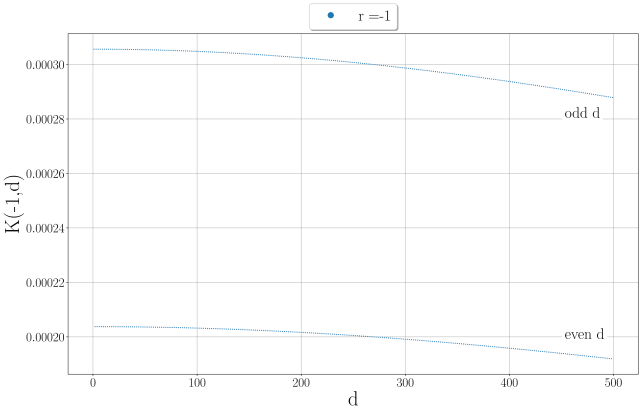


Figure 9: Behavior of $K(-1,d)$ when $n = 100$, $\varepsilon_{geo} = 0.2$

The top line indicates the behavior when d is odd and the bottom line when d is even. Both lines show a decrease wrt d. Hence either K(-1,1) or K(-1,2) will be the maximum value. Observe that K(-1,1) > K(-1,2).

Now let us examine in Figure 10 what happens for smaller values of r, keeping the same parameters for *Geo-Shuffle*.

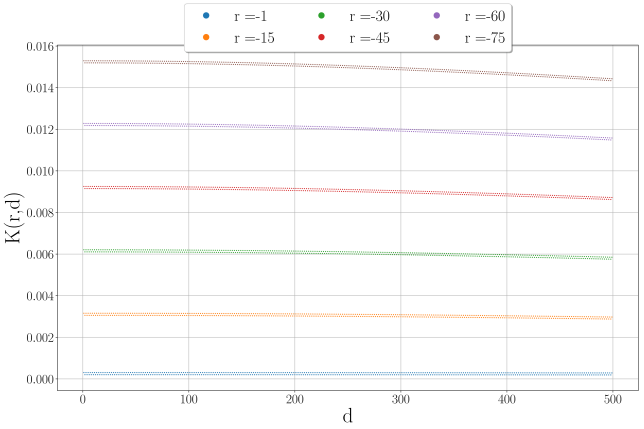


Figure 10: Behavior of $K(r,d)$, for multiple values of r when $n = 100$, $\varepsilon_{geo} = 0.2$

We can observe the same behavior as in the previous plot, $K(r,d)$ seems to be monotonically decreasing wrt d. Moreover let us increase ε_{geo} to 0.8 in Figure 11.

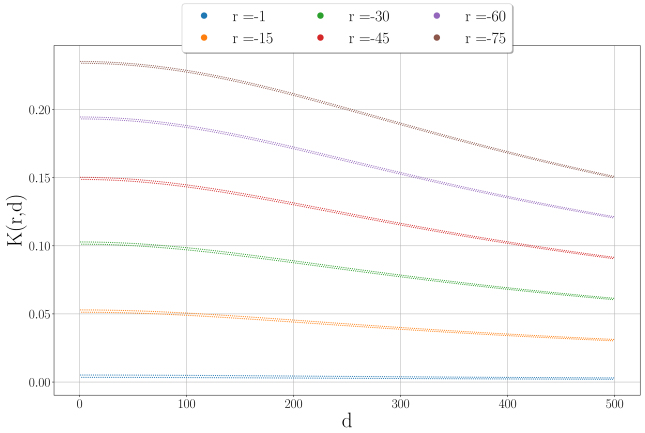


Figure 11: Behavior of $K(r,d)$, for multiple values of r when $n = 100$, $\varepsilon_{geo} = 0.8$

In this plot, our hypothesis is even more clear. Lastly, we also show in Figure 12 a plot with $n = 10000$. Here the lines are clearly more flat and the alleged behavior is less easily observable.

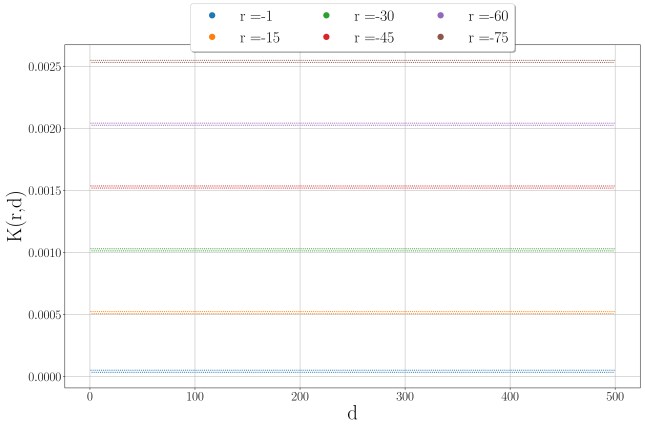


Figure 12: Behavior of $K(r,d)$, for multiple values of r when $n = 10000$, $\varepsilon_{geo} = 0.8$

Moreover, we would like to discuss this observation using the plots of Figure 2 and Figure 7. First in Figure 2 the PMF of SGDL is shown; we can observe that the ratio of probabilities between two points with distance d increases (at most) sub-exponentially. The natural logarithm of this ratio therefore grows slower than d. For instance let us consider the blue dotted line, which represents the case of SGDL(10,0.5). Observe that $P[SGDL(10,0.5) = 0] \approx 0.065$ and $P[SGDL(10,0.5) = 10] \approx 0.01$, making the natural logarithm of this ratio equal to ≈ 1.87, much smaller than the distance which is $d = 10$. The same observation can be made in the red dotted line where $P[SGDL(50,0.8) = 0] \approx 0.009$ and $P[SGDL(50,0.8) = 50] \approx 0.005$; the natural logarithm of this ratio is much smaller than $d = 50$.

Another useful plot is Figure 7; increasing d by 1 slightly increases the ratio. This increase is by itself smaller than the increase in d, even without considering its natural logarithm.

C Metric Privacy of the SGDL-Shuffle

Lemma C.1. *Let n users executing SGDL-Shuffle, with each user i sampling noise $N_i \sim SGDL(1/n, e^{-\varepsilon})$, with $\delta \in (0,1]$ and :*

$$c = \left\lceil \frac{u(1-e^{-\varepsilon})}{n} \right\rceil, \text{ where}$$
$$u = 1 + \frac{\sqrt{w(w-4)} - w}{2}$$
$$w = 2n \cdot \ln\left(\frac{1 - \sqrt[n]{1-\delta}}{2}\right)$$

Then:

$$1 - \prod_{i=1}^{n} \mathbb{P}\big[N_i \in [-c, c]\big] \leq \delta$$

Proof. Recall that the $SGDL(b,p)$ distribution is the difference between two Negative Binomial Distributions with parameters b and $1-p$ [28]. Observe that in SGDL, one Negative Binomial random variable bounds the positive noise and the other the negative noise. Hence it will be easier to find a tail bound for the Negative Binomial distribution.

If $Y \sim NB(b,q)$, we can directly apply a standard theorem [15] (p. 6) on the Negative Binomial Distribution to get, for any $u > 1$:

$$\mathbb{P}[Y > ubq] \leq \exp\left(\frac{-ub(1-1/u)^2}{2}\right) \tag{77}$$

But due to the relationship between SGDL and NB: $b = 1/n$ and $q = 1 - e^{-\varepsilon}$.

Let

$$B = \exp\left(\frac{-u(1-1/u)^2}{2n}\right) \tag{78}$$

If $c = \frac{u(1-e^{-\varepsilon})}{n}$, the probability that all n users sample noise in the interval $[-c, c]$ is $(1-2B)^n$.

Thus we need to show that:

$$\begin{aligned}
1 - (1-2B)^n &\leq \delta \\
1 - \delta &\leq (1-2B)^n \\
\sqrt[n]{1-\delta} - 1 &\leq -2B \\
B &\leq \frac{1 - \sqrt[n]{1-\delta}}{2} \\
\frac{-u(1-1/u)^2}{2n} &\leq \ln\left(\frac{1 - \sqrt[n]{1-\delta}}{2}\right) \\
-u(1 + 1/u^2 - 2/u) &\leq 2n \cdot \ln\left(\frac{1 - \sqrt[n]{1-\delta}}{2}\right) \\
-u - 1/u + 2 &\leq 2n \cdot \ln\left(\frac{1 - \sqrt[n]{1-\delta}}{2}\right)
\end{aligned}$$

Let:

$$w = 2n \cdot \ln\left(\frac{1 - \sqrt[n]{1-\delta}}{2}\right)$$

We need to solve the inequality:

$$u^2 + (w-2)u + 1 \geq 0$$

which yields:

$$1 - \frac{w + \sqrt{w(w-4)}}{2} \leq u \leq 1 + \frac{\sqrt{w(w-4)} - w}{2} \tag{79}$$

Because $u > 1$, we select:

$$u = 1 + \frac{\sqrt{w(w-4)} - w}{2} \tag{80}$$

Finally we use the ceiling function to ensure that c is an integer, as c will be used in our mechanism to represent a number of bits. □

Because we did not consider at all the input dataset:

Corollary 2. *The probability that at least one user of SGDL-Shuffle (ε, δ, n) truncates their input is at most δ, regardless of the input dataset.*

Now we can prove the privacy of SGDL-Shuffle:

Theorem 5.1. *For any $\delta \in (0,1]$ SGDL-Shuffle (ε, δ, n) is (ε, δ) - d_X - private.*

Proof. Since we have ensured, using δ, that every user reports a value in $[-c, c]$, we can just study the case when nobody has to truncate (calculating the upper bound probability i.e. without taking into account their secret x), using Proposition 3. We set H as the event that nobody truncates, K as the SGDL-Shuffle mechanism and M as the special case of SGDL-Shuffle where nobody truncates, $\delta_1 = \delta$ (Lemma C.1) and $\delta_2 = 0$. Note that H does not depend on the input dataset (Corollary 2).

We are going to argue that the total noise of SGDL-Shuffle, in that special no-truncation case, is the same as the one of the Geo-Central mechanism.

The analyst is able to view only the sum of the users' reported values as only a shuffled version of their bits is released to her (Proposition 1). This sum can be described by a random variable that follows $SGDL(1, e^{-\varepsilon})$. In turn, this is equal to the difference between two random variables sampled from the geometric distribution with parameter $1 - e^{-\varepsilon}$. From Claim B.1, in Section B, this is equal to Geo-Central (the geometric mechanism in the central model) which is ε metric private, as discussed in Section 4. Thus, the noise produced by the two mechanisms under this scenario (nobody truncating) can be described using the same distribution.

Using Proposition 3 we can conclude that the *SGDL-Shuffle* mechanism is (ε, δ) metric private in every case (somebody truncating or not).

□

C.1 SGDL in the local model

Let us restate the theorem:

Theorem 7.2. *SGDL-Shuffle(ε_S, δ, n), is (ε_L,δ) - d_x - private, in the local model, with:*

$$\varepsilon_L = \max_{d_x} \frac{\ln(g(0, d_x(x,x'), e^{-\varepsilon_S}, 1/n))}{d_x(x,x')} \tag{8}$$

for the function g as defined in Theorem 4.1.

Proof. We are going to use the same approach as in Theorem B.1.

Consider two elements x and x' that belong to a dataset X with distance $d_x(x,x') = d$. It will be more convenient to assume, w.l.o.g. that for any $r \in \mathbb{Z}$ s.t. r is the central point in the interval $[x,x']$: $x = \lfloor r - \frac{d}{2} \rfloor$ and $x' = \lfloor r + \frac{d}{2} \rfloor$. Note that we have to use the floor function since we consider datasets which contain only integers.

Recall the Probability Mass Function of the SGDL distribution:

$$\mathbb{P}(Y = m) = (1-p)^{2\cdot\beta} \sum_{k=|m|}^{\infty} \binom{\beta+k-1}{k} \binom{\beta+k-|m|-1}{k-|m|} p^k p^{k-|m|}$$

Moreover, recall the function $h(r,d)$ from Theorem B.1:

$$h(r,d) = \frac{\mathbb{P}[\lfloor r - \frac{d}{2} \rfloor = SGDL(\beta,p)]}{\mathbb{P}[\lfloor r + \frac{d}{2} \rfloor = SGDL(\beta,p)]} = \frac{\sum_{k=|\lfloor r-\frac{d}{2} \rfloor|}^{\infty} \binom{\beta+k-1}{k} \binom{\beta+k-|\lfloor r-\frac{d}{2} \rfloor|-1}{k-|\lfloor r-\frac{d}{2} \rfloor|} p^k p^{k-|\lfloor r-\frac{d}{2} \rfloor|}}{\sum_{k=\lfloor |r+\frac{d}{2}| \rfloor}^{\infty} \binom{\beta+k-1}{k} \binom{\beta+k-\lfloor |r+\frac{d}{2}| \rfloor-1}{k-\lfloor |r+\frac{d}{2}| \rfloor} p^k p^{k-\lfloor |r+\frac{d}{2}| \rfloor}} \tag{81}$$

Note that the function h also depends on β and p (where in SGDL-Shuffle $\beta = 1/n$ and $p = e^{-\varepsilon_S}$), which we omit for notational convenience. For the remainder of this section we will hence write $h(r, d)$ and not $h(r, d, p, \beta)$.

In order to prove metric privacy, we have to bound the ratio of probabilities to observe two events that differ by d. Hence, we would like to find the values of ε_L that satisfy the following inequality:

$$(81) \leq e^{\varepsilon_L \cdot d} \tag{82}$$

Note that the inverse ratio should also be calculated, because, depending on d, it could be larger. We are looking to find an upper bound, hence we are interested in the maximum of these two ratios.

Let:

$$g(r, d) = max\{h(r, d), \frac{1}{h(r, d)}\} \tag{83}$$

We now need to find a value for r and d that maximize $\varepsilon_L = \frac{\ln(g(r,d))}{d}$.

Note that g depends on h which in turn depends on the parameters r, d, β, p. Since we have set $\beta = 1/n$ and $p = e^{-\varepsilon_S}$, we are only interesting in finding the proper values for r, d. Hence we omit writing $g(r, d, p, \beta)$ and we use the simpler form of $g(r, d)$. However we retain the form $g(r, d, p, \beta)$ in the statement of Theorem 7.2 in order to help the reader understand the relationship between the variables without referring to the proof.

In Theorem B.1, recall that $|g(r, d)|$ was monotonically increasing function wrt r, therefore, it was necessary to utilize δ in order to find r.

However now the choice is simpler, since the fact that $\beta < 1$, makes $|g(r, d)|$ a monotonically decreasing function wrt r (Figure 13 and Figure 14); we can simply take $r = 0$ as the maximum. In fact, $g(r, d)$ has the same value for every $r \in \{-d, \ldots, 0\}$. Proof is analogous to the proof of Theorem B.1.

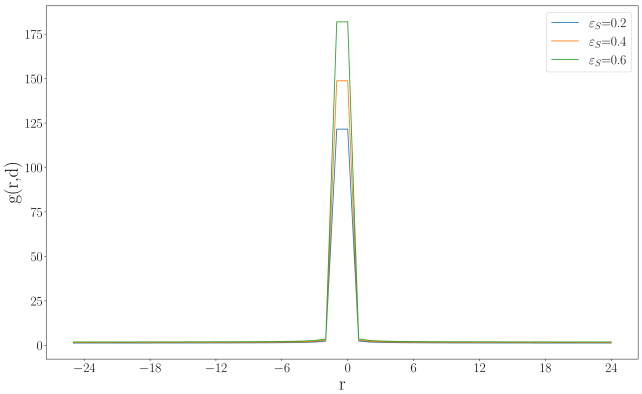


Figure 13: $g(r, d)$ function for $\beta = 0.01$ when $d = 1$

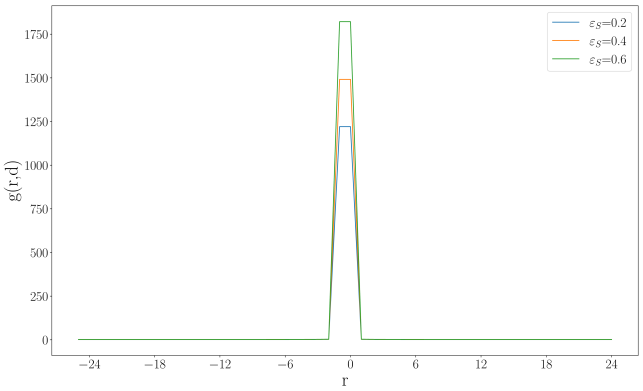


Figure 14: $g(r, d)$ function for $\beta = 0.001$ when $d = 1$

Similarly to Geo-Shuffle, one might directly take $d_x = 1$ and calculate:

$$\varepsilon_L = \ln(g(0, 1, e^{-\varepsilon_S}, 1/n)) \tag{84}$$

□

D Compositionality

The output of the location data experiments of Section 6 will be a composition of two mechanisms, one for each dimension. The following theorem allows us to find the privacy of the composition:

Theorem D.1. *Let mechanisms $M_1, M_2 : \mathbb{R} \to \mathbb{R}$ be (ε, δ)-$d_{\mathbb{R}}$-private. Then their composition $M = M_1 \times M_2 : \mathbb{R}^2 \to \mathbb{R}^2$ is $(\varepsilon, 2\delta)$-$||\cdot||_1$-private. In other words, for all locations $x, x' \in \mathbb{R}^2$ and $S \subseteq \mathbb{R}^2$ we have:*

$$\mathbb{P}[M(x) \in S] \leq e^{\varepsilon ||x - x'||_1} \mathbb{P}[M(x') \in S] + 2\delta\,. \tag{85}$$

Proof. Let us simplify the notation and denote:

$$P_i = \mathbb{P}_{\mathcal{M}_i}[y_i \in S_i | x_i]$$
$$P'_i = \mathbb{P}_{\mathcal{M}_i}[y_i \in S_i | x'_i]$$

for $i \in \{1, 2\}$. As mechanisms $\mathcal{M}_1$ and $\mathcal{M}_2$ are applied independently, we have:

$$\mathbb{P}_{\mathcal{M}_1,\mathcal{M}_2}[(y_1, y_2) \in S_1 \times S_2 | (x_1, x_2)] = P_1.P_2 \tag{86}$$
$$\mathbb{P}_{\mathcal{M}_1,\mathcal{M}_2}[(y_1, y_2) \in S_1 \times S_2 | (x'_1, x'_2)] = P'_1.P'_2 \tag{87}$$

Therefore, we obtain:

$$\begin{aligned}
\mathbb{P}_{\mathcal{M}_1,\mathcal{M}_2}[(y_1, y_2) \in S_1 \times S_2 | (x_1, x_2)] &= P_1.P_2 \\
&\leq \left(\min\left(1 - \delta, e^{\varepsilon\, d(x_1, x'_1)} P'_1\right) + \delta\right) \\
&\times \left(\min\left(1 - \delta, e^{\varepsilon\, d(x_2, x'_2)} P'_2\right) + \delta\right) \\
&\leq m_1 m_2 + \delta m_2 + m_1 \delta + \delta^2 \\
&\left[\text{where } m_i = \min\left(1 - \delta, e^{\varepsilon\, d(x_i, x'_i)} P'_i\right)\right] \\
&\leq e^{\varepsilon\, d(x_1, x'_1)} P'_1\, e^{\varepsilon\, d(x_2, x'_2)} P'_2 \\
&+\delta(1 - \delta) + (1 - \delta)\delta + \delta^2 \\
&= e^{\varepsilon\, d(x_1, x'_1) + \varepsilon\, d(x_2, x'_2)} P'_1 P'_2 \\
&+\delta - \delta^2 + \delta - \delta^2 + \delta^2 \\
&\leq e^{\varepsilon ||x - x'||_1} \\
&\times \mathbb{P}_{\mathcal{M}_1,\mathcal{M}_2}[(y_1, y_2) \in S_1 \times S_2 | (x'_1, x'_2)] \\
&+2\delta
\end{aligned} \tag{88}$$

□

Note that the results of Sections 3, 4.1 and 5 gives us privacy wrt $||\cdot||_1$. However, the standard notion of geo-indistinguishability considers $||\cdot||_2$. The well-known inequality $||x||_1 \leq ||x||_2 \cdot \sqrt{2}, x \in \mathbb{R}^2$, however, gives us the following result:

Proposition 7. *If $M : \mathbb{R}^2 \to \mathbb{R}^2$ satisfies (ε, δ) - $||\cdot||_1$ - privacy then it also satisfies $(\varepsilon\sqrt{2}, \delta)$ - $||\cdot||_2$ - privacy.*

Using the above results, to achieve (ε, δ) - $||\cdot||_2$ - privacy, we can apply any mechanism to each dimension with $\frac{\varepsilon}{\sqrt{2}}$ and $\frac{\delta}{2}$.

E Additional Experiments

In this section we include additional experiments with varying parameters and utility metrics, which are missing from Section 6.

E.1 First experiment: Synthetic Data

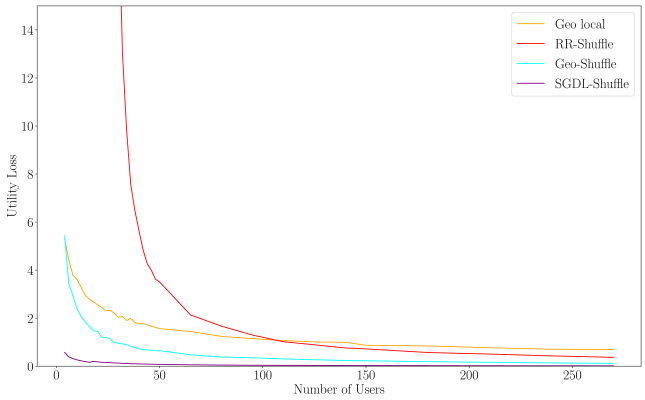


Figure 15: Experiment 1: $\varepsilon = 0.1$, $\delta = 10^{-4}$, Utility Metric: Mean Absolute Error

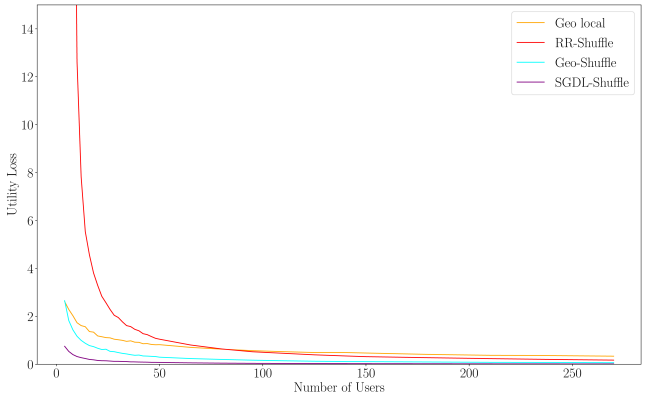


Figure 16: Experiment 1: $\varepsilon = 0.2$, $\delta = 10^{-3}$, Utility Metric: Mean Absolute Error

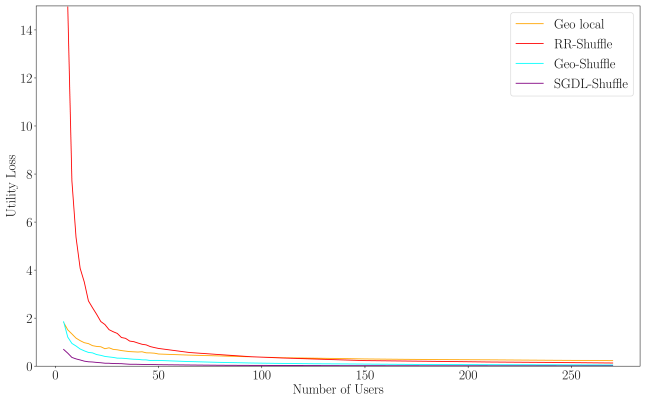


Figure 17: Experiment 1: $\varepsilon = 0.3$, $\delta = 10^{-4}$, Utility Metric: Mean Absolute Error

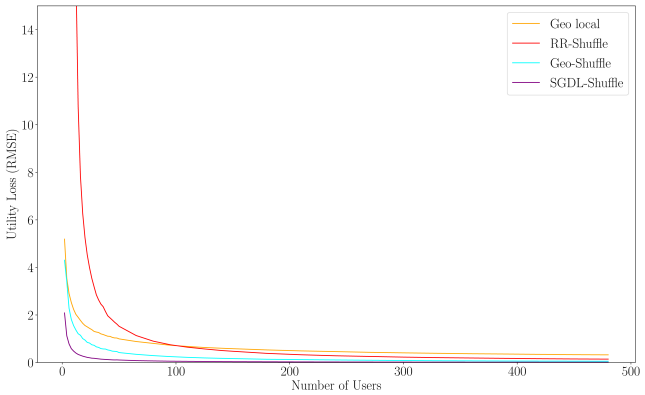


Figure 18: Experiment 1: $\varepsilon = 0.2$, $\delta = 10^{-4}$, Utility Metric: Root Mean Square Error

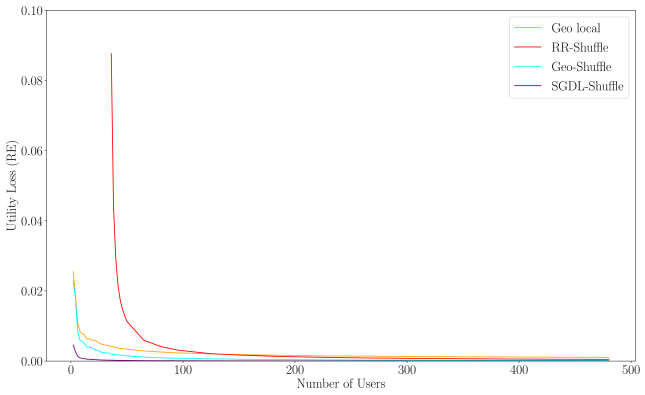


Figure 19: Experiment 1: $\varepsilon = 0.1$, $\delta = 10^{-4}$, Utility Metric: Relative Error

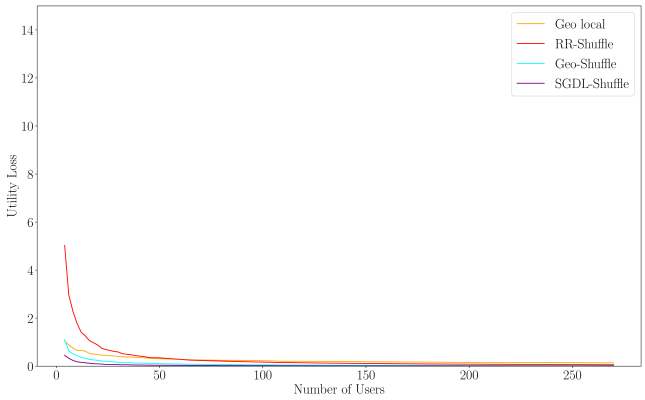


Figure 20: Experiment 1: $\varepsilon = 0.5$, $\delta = 10^{-2}$, Utility Metric: Mean Absolute Error

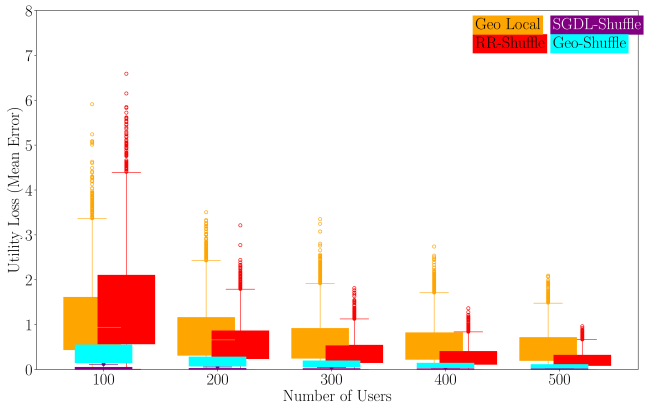


Figure 21: Experiment 1: $\varepsilon = 0.2$, $\delta = 10^{-4}$, Boxplot , Utility Metric: Mean Absolute Error

E.2 Second experiment: Location Data

First, we show in Figure 22 a map of the addresses (blue dots) that we used on the location data experiment. Observe that the addresses are concentrated in the center of the city. We repeat the experiment using only the addresses of those who live in the suburbs [6]. We therefore name this dataset *Dispersed Data*.

[6]We omit users whose latitude $\in [30.16737, 30.45908]$ and longitude $\in [-97.877248, -97.627248]$

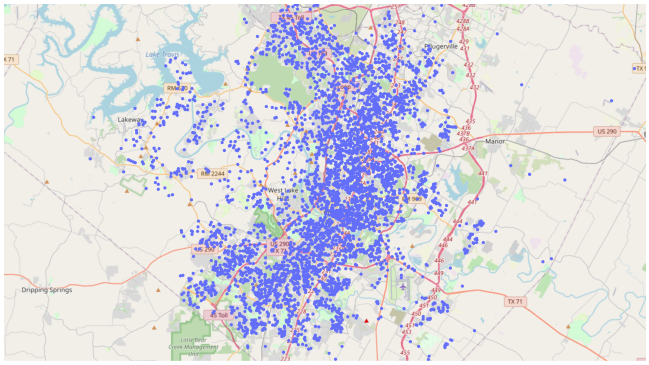


Figure 22: Map of addresses (blue dots) in Austin, Texas.

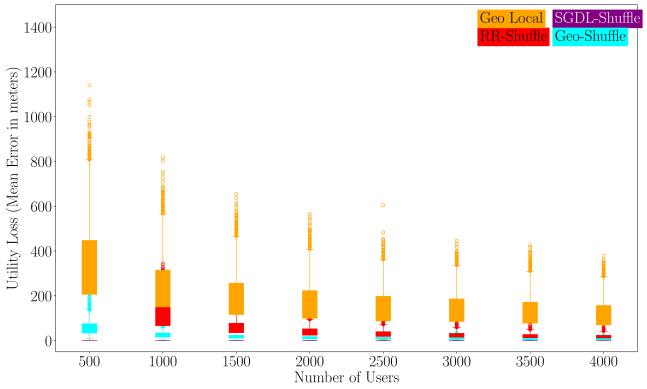


Figure 23: Experiment 2: $\varepsilon = 0.1$, $\delta = 10^{-4}$, size of grid 1000x1000 squares and privacy radius of $r = 600$ meters. Utility Metric: Euclidean Distance

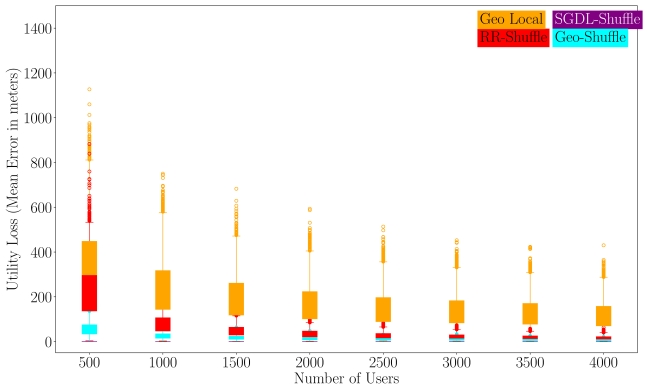


Figure 24: Experiment 2: $\varepsilon = 0.2$, $\delta = 10^{-4}$, size of grid 500x500 squares and privacy radius of $r = 600$ meters. Utility Metric: Euclidean Distance

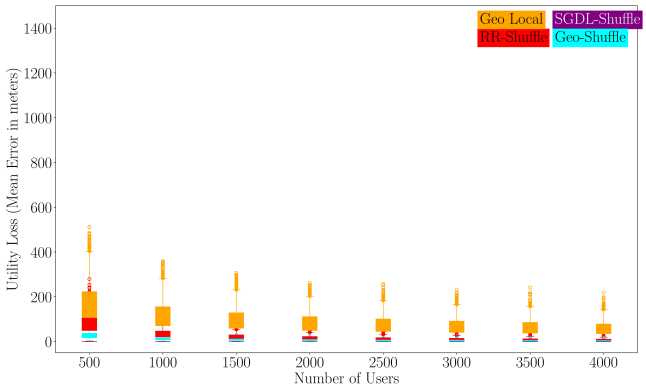


Figure 25: Experiment 2: $\varepsilon = 0.3$, $\delta = 10^{-4}$, size of grid 1500x1500 squares and privacy radius of $r = 500$ meters. Utility Metric: Euclidean Distance

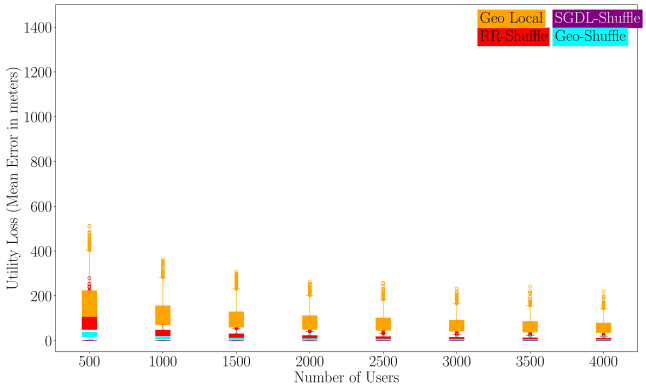


Figure 26: Experiment 2: $\varepsilon = 0.2$, $\delta = 10^{-4}$, size of grid 1000x1000 squares and privacy radius of $r = 300$ meters. Dispersed Data. Utility Metric: Euclidean Distance

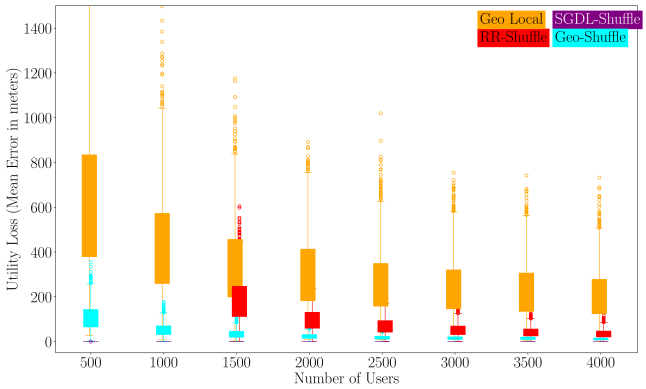


Figure 27: Experiment 2: $\varepsilon = 0.1$, $\delta = 10^{-4}$, size of grid 1000x1000 squares and privacy radius of $r = 600$ meters. Utility Metric: Manhattan Distance

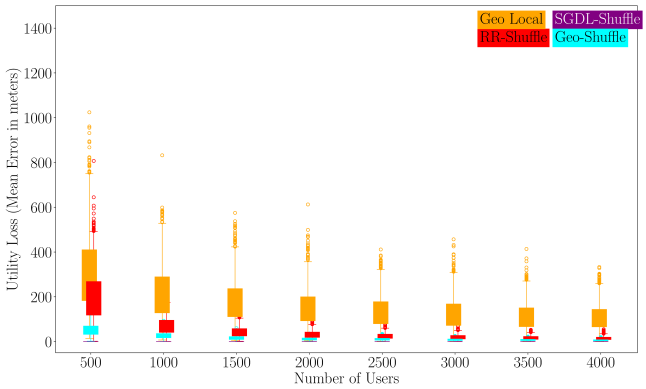


Figure 28: Experiment 2: $\varepsilon = 0.2$, $\delta = 10^{-4}$, size of grid 1000x1000 squares and privacy radius of $r = 600$ meters. Dispersed Data. Utility Metric: Manhattan Distance

F Concentration Inequalities

Theorem F.1 (Chernoff bound). *If $x_1, \ldots, x_n$ are independent {0,1}-valued random variables, each with mean μ, then, for every $\beta > 0$,*

$$\mathbb{P}[\mu n - \sum x_i < \sqrt{2\mu n \log \tfrac{1}{\beta}}] \geq 1 - \beta \text{ and}$$
$$\mathbb{P}[\sum x_i - \mu n < \sqrt{3\mu n \log \tfrac{1}{\beta}}] \geq 1 - \beta$$

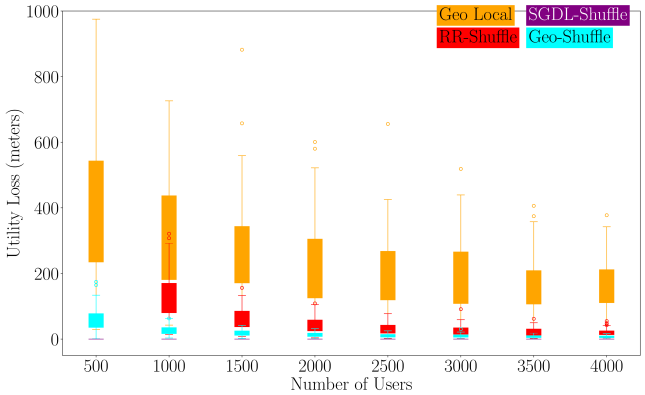


Figure 29: Experiment 2: $\varepsilon = 0.15$, $\delta = 10^{-4}$, size of grid 1000x1000 squares and a privacy radius of $r = 600$ meters. Dispersed Data. Utility Metric: Euclidean Distance

Theorem F.2 (Hoeffding's inequality). *If $x_1, \ldots, x_n$ are independent random variables, each with mean μ and bounded in (a, b), then, for every $\beta > 0$,*

$$\mathbb{P}\left[|\sum x_i - \mu n| < (b-a)\sqrt{\tfrac{1}{2} n log \tfrac{2}{\beta}}\right] > 1 - \beta$$

Theorem F.3 (Bernstein's inequality). *If $x_1, \ldots, x_n$ are independent random variables, each with mean 0. bounded in $[-1, 1]$, and $\sigma^2 > \frac{4}{9n} \log \frac{2}{\beta}$ then for every $\beta > 0$:*

$$\mathbb{P}\left[|\sum x_i| < 2\sigma\sqrt{n \log \tfrac{2}{\beta}}\right] > 1 - \beta$$

G Communication Cost

Here we explore the idea to use a bigger quantization value Q. We present in Algorithm 8 a version of this approach [7]. The algorithm performs a euclidean

[7] A similar approach for encoding bounded real values has been studied in [11].

division of x (the number to be encoded) with the granularity value Q getting the quotient x_q and the remainder x_r. Then it outputs $x_q + Ber(x_r/Q)$ ones, where the Bernoulli distribution is used to probabilistically round the output.

Algorithm 8: $\mathcal{U}_Q(x, r)$: Unary encoding of x in r bits under granularity Q

Input : $x \in \mathbb{N}$, $Q \in \mathbb{N}$, $r \in \mathbb{N}$, where $x \leq r \cdot Q$
Output: $(b_1, \ldots, b_r) \in \{0,1\}^r$
$x_q \leftarrow x//Q$ (Quotient)
$x_r \leftarrow x \mod Q$ (Remainder)
for $j = 1, \ldots, r$ **do**

$$\mathbf{b}_j = \begin{cases} 1 & j \leq x_q \\ \text{Ber}(x_r/Q) & j = x_q + 1 \\ 0 & j > x_q + 1 \end{cases}$$

end
Return $(b_1, \ldots, b_r)$

This new unary encoding can substitute Algorithm 1 in the above mechanisms and the analyst can simply multiply the output by Q to get the final result. If we assume a reasonably large r (the size of the encoded vector), then we can observe that summing the bits of the output of Algorithm 8 and multiplying them by Q is an unbiased estimator of the initial value x:

$$\mathbb{E}[\,Q(x_q + Ber(x_r/Q))\,] = Q \cdot x_q + x_r = x$$

This approach however will decrease the metric privacy of the proposed mechanisms. To see why, recall the example that we discussed in 4.1. Let us take the example of a dataset where everybody holds a small value, say x_s, but one outlier has a particular large value, say x_L. Recall that applying an obfuscation mechanism (which is metric private) will produce values close to the initial ones with higher probability (unlike standard LDP). However, if Q is the quantization value, setting $Q = x_L$will make the user that holds x_L to report 1 and every other user to report 0 (coming from $Ber(x_s/x_L)$ of Algorithm 8) with very high probability. After shuffling these bits and releasing them to the analyst, the analyst can confidently assume that the bit "1" corresponds to someone actually having a large value close to x_L.