Optimizing Differential Privacy in Federated Analytics under Known Input Distributions

Ferran Alborch
EURECOM
Orange Innovation
Sophia Antipolis / Caen, France
ferran.alborch@eurecom.fr

Andreas Athanasiou
TU Delft
INRIA & École Polytechnique
Delft, Netherlands / Palaiseau, France
a.athanasiou@tudelft.nl

Pascal Reisert
Institute of Information Security
University of Stuttgart
Stuttgart, Germany
pascal.reisert@sec.uni-stuttgart.de

***Abstract*—Differential privacy (DP) is one of the most efficient tools for protecting the privacy of individual data holders under computation. This property guarantees that the computation outputs for every pair of adjacent input sets are statistically indistinguishable with respect to a given parameter ε, which is independent of the likelihood that specific inputs occur or not. While the distribution of input sets is generally unknown, in some use cases (approximate) information about it might be available. If the latter is the case, two adjacent inputs of one individual are sometimes already obfuscated by other inputs and the computation itself (i.e., without any additional noise). For example, if the sum of n independent and identically distributed uniformly random bits outputs approximately $n/2$, both values for the first bit remain (almost) equally likely for large n.**

Based on this observation, we present a new DP mechanism that uses an estimate of the input distribution to reduce the noise addition (compared to standard DP) and hence improves the accuracy of the output. We first explore this idea in the central model, where a single central party collects all data. Then, we provide a new technique (possibly of independent interest) that allows multiple entities to jointly generate reduced noise, using the property of infinite divisibility. This allows each party to individually add noise to their respective inputs, e.g., in Federated Analytics applications.

We apply our theoretical results, both for the single and multi-party setups, to perform data analysis over human resources data from different subsidiaries within a corporate group. Our benchmarks show that our new DP mechanism provides more accurate outputs while retaining the same privacy level as state-of-the-art DP approaches using the geometric mechanism.

***Index Terms*—Differential Privacy, Federated Analytics, Multi-Party Computation**

I. Introduction

In recent years, there has been a continuous increase in the need to protect sensitive private data. In Federated Analytics [1], multiple parties, each with its own local dataset, work collaboratively under the coordination of a central entity to produce a joint statistical result and release it to an analyst. Since the analyst might be malicious, each party does not wish to release its raw dataset and thus the output statistic needs to be obfuscated. A local dataset can contain data from various sources, and we assume that the original data owners trust the corresponding party and provided their data freely or by contractual agreement. We therefore call the party that manages the local data set a locally trusted entity (LTE). Generally, each data owner only trusts a single LTE. The LTE runs the actual protocol with other LTEs and the central entity to produce the overall obfuscated statistic. This computation can include one (or multiple) rounds of local computations followed by interactive rounds that ultimately provide the output to the analyst.

In this paper, we study this setup (cf. Section IV) and develop new privacy-preserving mechanisms that provide better utility than the established solutions while maintaining the same level of privacy for the individual data owners.

Before we explain our new mechanism, we want to present some real-world use cases where our setup applies. Consider first a (holding) company with multiple subsidiaries that seeks to collect and analyze human resources (HR) data from its employees. This data is often already collected by the respective subsidiaries, e.g., their HR department. Each employee trusts (or has to trust) their subsidiary (i.e., an LTE) but not the other subsidiaries or the holding company. Each LTE needs to protect its employees' privacy against the holding company (and the other LTEs), but also needs to provide statistical data to the holding company. We will use this use case throughout the paper, since we later in Section VII evaluate our new mechanism using data from a real-world corporate group.

Another example of our setup are inter-entity bank transfers. In this use case, some controlling entity wants to track down fraud by analyzing transfers between different banks. Each client trusts their bank (i.e., an LTE) but not necessarily any other, nor the controlling entity. Similarly to before, the LTEs are obliged to report transfers to the controlling entity while needing to maintain the privacy of their clients.

Federated Learning (FL) [2] delivers a whole set of applications of our setup, depending for what it is used. In FL different clients (LTEs) train (possibly over many epochs) their local machine learning (ML) models over often sensitive end-user data. These local models are aggregated to a global output model by a central entity. While FL aligns with the aforementioned setup, i.e., data owners (e.g., patients) trust their LTE (e.g., hospitals) but no other entities, the (typically) multiple epochs in FL pose additional challenges which we discuss in Section VIII.

In addition to the same setup, the described use cases also share a common interactive phase, where the parties aggregate the results of the local precomputations, i.e., the local HR-

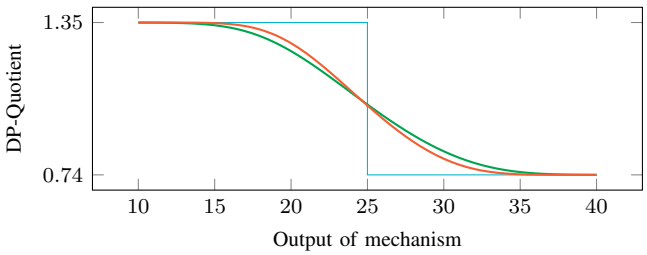

Fig. 1. Smoothed DP-Quotients (cf. Remark 1) for the distributions $\mathsf{Geo}_{0.3}$ (blue), $\mathsf{Bin}(0.5, 50) * \mathsf{Geo}_{0.3}$ (green) and $\mu_1 * \mathsf{Geo}_{0.3}$ (red), where μ_1 is more concentrated than the binomial distribution (similar to Fig. 2). For $\varepsilon = 0.3, q = e^{-\varepsilon} \approx 0.74, q^{-1} \approx 1.34$. Bin denotes a binomial and Geo a geometric distribution (cf. Definition 4).

statistic in the corporate use case, the number of bank transfers in the banking use case or the local ML models in FL. We hence concentrate in our setup on integer summation. Note that the integer summation problem is also the building block for more complex queries and many FL aggregation functions, such as the widely adopted FedAvg [2].

A natural solution to the privacy challenges in our setup is provided by Differential Privacy (DP) [3], [4]. DP is regularly used in applications that fall within our setup, e.g., for privacy-preserving FL [5]. Differential Privacy provides a formal way to measure the privacy loss of a mechanism, i.e., how much information about the inputs can (at worst) still be deduced from its output. In scenarios where multiple independent users contribute their own data, DP can be applied either locally (using the *local model*) or centrally (using the *central model*).

In the local model of DP [6], each user injects noise into its data. The collection of these individually obfuscated data samples is then published. It no longer leaks significant information about the single user's data, but also comes with low utility compared to other DP-approaches. In the central model of DP, a trusted curator collects the raw data and injects noise to obfuscate the result. While the central model comes with high utility, it assumes a trusted third party, the curator, which might not be realistic, since real-world curators may, for example, be vulnerable to cyberattacks, leading to data leaks.

A classical alternative to a trusted third party is secure Multi-Party Computation (MPC) [7]. In an MPC protocol, several parties, with their own private input data, compute a previously agreed upon function over the ensemble of inputs while leaking nothing else than the result of the function. In the context of DP, MPC has been widely studied [8]–[11] and is used in the central model to generate the DP noise in a distributed manner. Unfortunately, MPC protocols usually come with an overhead in runtime and communication compared to the trusted third-party setup. Additionally, their efficiency decreases with the complexity of the evaluated function. While the simple aggregation[1] or shuffling often used in DP can be realized efficiently in MPC, sampling noise according to a predefined distribution is comparably inefficient [9], [12]. In DP, it is therefore most efficient to generate the necessary noise locally for each party and then to enter the already obfuscated results into an MPC protocol, e.g., for aggregation, which guarantees only the output is released to the analyst, e.g., our holding company.

This approach works particularly well with *infinitely divisible noise distributions* [9], [13], [14]. If a noise distribution σ is infinitely divisible, noise can be generated by the users in a way that the sum of these noises is described by σ. Hence, each party has to add only a small part of the noise, and the secure aggregation ensures that the analyst only receives the final result obfuscated with noise from σ. In contrast to the local model, with MPC the obfuscated data from a single LTE never becomes visible and, therefore, requires less obfuscation.

For our setup, we could use both the local and central variants of DP. With local DP, each LTE could add noise to each employee's data sample, which would lead to the aforementioned utility issues. Besides, since LTEs are trusted, employing purely local DP would not be optimal. Hence, in this work, we focus on central DP in the sense that the LTE first produces a statistic for its employees, then obfuscates the statistic, and the MPC protocol (run among the LTEs) provides the holding company with the aggregated statistic. Note that in the central approach, we add noise to the result of a precomputation, e.g., the average e-mail number in a group of employees of the respective LTE. We do not add noise to a single data sample like in local DP. In particular, if we use a standard mechanism for noise sampling, like the Laplace or Geometric mechanism [3], it guarantees that *two outputs of the precomputation* can no longer be distinguished (up to ε-DP). However, our goal in this paper is to protect the *privacy of individual data owners*. Depending on the actual precomputation, its result might already obfuscate the individual data sample partially or even completely, such that noise addition is no longer required.

The obfuscating effect of precomputations or (non-injective) functions in general is well-known and often used in privacy. For example, for sufficiently large elections (with not too many, too unlikely choices), we expect the election tally to hide or reveal (almost) nothing about the choice of a single voter [15]. In this paper, we want to study the effect of this inherent obfuscation on DP. We present a new mechanism that reduces the (artificial) noise added by each LTE depending on the already existing obfuscation. As a result, our new mechanism produces more accurate results than state-of-the-art approaches based on Laplacian or geometric noise, using an estimation of the distribution over the secrets.

Our New DP-Mechanisms

While this inherent obfuscation by a precomputation might not be surprising, finding suitable DP mechanisms that use this phenomenon poses several technical difficulties, which might be the reason why this approach has not been discussed in the literature. We want to discuss these difficulties and our solutions briefly for the aforementioned use case.

[1]We will focus on aggregation in the following, but other simple operations like averaging or shuffling work analogously.

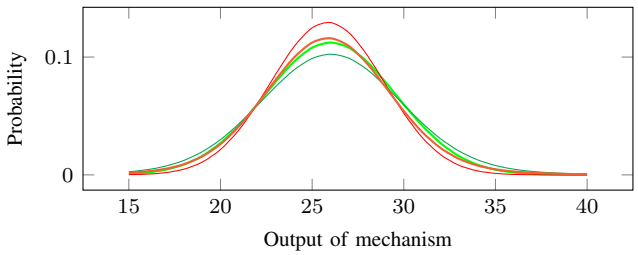

Fig. 2. Smoothed probability distributions $f_1 \circ \chi_1 = \mathsf{Bin}(0.5, 50)$ (black) and $f_1 \circ \chi_1 * \mathsf{SGDL}_{1/4}$ (green). The distribution μ_1 (red) is constructed such that $\mu_1 * \mathsf{SGDL}_{1/4}$ (blue) approximates $f_1 \circ \chi_1$ (cf. Lemma 2 for SGDL.)

Recall that the geometric mechanism with a parameter $q = e^{\varepsilon}$ provides $\varepsilon-$DP (cf. Definition 1). Even more, if two statistical outputs x, y differ by 1, the DP-quotient, i.e., the relation between the likelihood that an obfuscated result was created from x or y, is always exactly q or q^{-1} (see Fig. 1, blue line). Note that the geometric mechanism offers the best accuracy for the integer summation problem among noise generation mechanisms that support infinite divisibility [13].

First, let us consider the simple setup of a single LTE which computes the overall number of e-mails by its employees. We model that sending an e-mail or not is a simple Bernoulli experiment. In other words, their sum is described by a binomial distribution. After the LTE has gathered the e-mails, geometric noise is added to the sum. We are interested in how much information an adversary can gain about the result of a single Bernoulli experiment given access to the obfuscated sum. We see in Fig. 1 that the DP-quotient for this mechanism takes values strictly between q and q^{-1}. In particular, less information about the number of e-mails of a specific employee can be deduced if the other employees send an average amount of e-mails. Note that the LTE sees the raw overall number of e-mails (i.e., not the obfuscated sum), and can adapt its noise to this value. Hence, the LTE can add less noise (compared to the geometric mechanism) if it sees a sum close to the mode of the binomial distribution and the standard geometric noise for outliers. With this technique, we build in Section V a mechanism that is never worse than the geometric mechanism and whose accuracy loss converges to 0 if the number of employees becomes large.

Note that we assume the LTE to have access to the overall distribution of e-mails (sent by its employees) or an approximation thereof. This contrasts with classical DP solutions, which do not make such an assumption. Given our setup, this is a likely scenario since HR departments should have access to earlier statistics and can furthermore use a Bayesian approach [16] to improve their approximation. Additionally, information about e-mail-sending behavior in companies or in general might be available and can also be used to approximate the distribution. A similar assumption also holds for the other use cases, e.g., banks usually have an accurate knowledge of their daily transfers and in FL approximate knowledge of the input distribution (called *shadow distribution*) is often assumed [17]. Our approach provides ε-DP as long as the LTE does not overestimate the obfuscating effect of its distribution—an underestimate leads to less utility gains compared to standard solutions, but not to any leakage (cf. Section VI-E).

We now turn to the actual multi-LTE case. Recall that (discrete symmetric) geometric noise is infinitely divisible, and its decomposition follows symmetric generalized discrete Laplacian noise [13], [18]. Unfortunately, the property of infinite divisibility does not transfer to our reduced noise distribution from the single LTE case, i.e., we cannot simply split the reduced noise mechanism of Section V. Even worse, if one LTE modifies its noise distribution close to the mode as in the single-party case, this modification is spread under convolution when the obfuscated results of the LTEs are added up (e.g., in an MPC protocol). We were not able to control the behavior of the aggregated noise and prove privacy with the analog of the single-LTE approach.

Instead, we chose a different approach. Similar to Fig. 1, we start by making the privacy condition tight, i.e., providing the same level of privacy for all employees. We can see in Fig. 1 that the privacy conditions become tighter if the (approximated) distribution of e-mails is concentrated. In the most extreme case, we have a point function, and the DP-quotient becomes a step function (e.g., $\mathsf{Geo}_{0.3}$ in Fig. 1). While the LTE cannot change the (approximated) distribution χ of e-mails, it can construct a more concentrated distribution μ itself. Our mechanism then works as follows: The LTE gets the sum (approximately) distributed as χ. It then uses an algorithm A to map the sum x to some value y where A is constructed such that $y \sim \mu$. Then, the LTE samples a Laplacian error $e \leftarrow \mathsf{SGDL}$ (where SGDL notates the symmetric generalized discrete Laplace distribution, cf. Lemma 2), adds it to y, and outputs the result $y + e$ if it is very big or very small. If $y + e$ is moderately sized, i.e., within a finite interval I, a finite distribution τ_x on I for each x is used to sample the output.

We show that accuracy (in terms of the mean absolute error (MAE)) is not affected for very big or very small $y + e$, i.e., the output of our mechanism then has the same error as the standard Laplacian error applied to x. Inside I, we can optimize the finite distribution τ_x such that it minimizes the MAE while satisfying the privacy constraints and such that the output of our mechanism is distributed as $\mu * \mathsf{SGDL}$ (if we vary overall x). The Laplacian parts SGDL of the different LTEs add up nicely under aggregation to a geometric noise. The resulting overall distribution is more concentrated than with the Laplacian/geometric approach since μ (for each LTE) is more concentrated than χ (for that LTE). Hence, the privacy level is closer to q or q^{-1}. The higher concentration of μ (cf. Fig. 2) around the mode also allows τ_x to output more accurate values. We refer to Section VI for the detailed construction.

Finally, we apply our results in the single and multi-party setup to aggregate HR data from different LTEs within a corporate group. Our evaluation shows that our DP-mechanism leads to more accurate outputs than the state-of-the-art geometric mechanism while retaining the same privacy level.

Contributions

- We present a new DP-mechanism that uses the obfuscation inherent in precomputations to reduce the noise addition and improve accuracy compared to the state-of-the-art geometric mechanism when the input distribution is known.
- In the case of a single party, the accuracy loss of our mechanism is (almost) minimal for large datasets. For our summation queries, it converges to 0.
- For the multi-party case, we develop a new technique to construct (infinitely) divisible mechanisms (Section VI).
- We evaluated our mechanism both for generic benchmarks and on a human resource use-case [19]. Our benchmarks show that our new DP-mechanism outperforms the geometric mechanism in terms of accuracy while providing the same level of privacy (Section VII).

II. Related Work

In this section, we discuss related works that can be applied in the setup outlined in Section I: a distrusted central entity that wants to analyze private users' data, with possibly some intermediate local entities that can only access parts of the data. We already introduced in Section I the local model, where each user obfuscates their inputs themselves. This setup does not need intermediate entities, i.e., our LTEs, but generally comes with poor utility. An improvement of the local model with respect to utility is the *shuffle model* [12], [20], [21]. The shuffle model uses an intermediate honest shuffler that mixes the already obfuscated data from the users. Since shuffling breaks the link between the owner and their value, the users now need to add less noise to reach the same privacy level as in the local model (i.e., without a shuffler), and thus, utility is improved. The shuffler can either be a trusted third party (e.g., using trusted hardware) or realized by cryptographic protocols, e.g., MPC protocols [22]–[24] or MixNets [25], [26]. However, the MPC approach usually suffers from low efficiency for a large number of users.

For the integer summation problem considered in this work, the shuffler can also be replaced by a simple aggregator, who outputs the sum of the inputs the users provide. The MPC protocols to realize aggregators, e.g., additive secret sharing-based protocols like [27], are generally more efficient than shuffling protocols [11]; they scale constantly in the number of parties. While shufflers can also be used for other Federated Analytics tasks, e.g., heavy hitters [28] and private set unions [29], for our use case, an aggregator is enough (cf. Section IV).

In this work, we assume that the (approximate) input distribution is known to the LTEs. Variants of DP have been proposed for this scenario (e.g., noiseless privacy [30]), providing specific privacy guarantees when the secrets follow a particular distribution. In contrast, we do not restrict to any specific distribution. Moreover, smooth sensitivity [31] calculates the exact sensitivity of the DP query on the given dataset, thereby reducing the amount of added noise. Similarly to our approach, it requires (approximate) knowledge of the input distribution. However, we further exploit this knowledge by proposing mechanisms that optimize their noise based on the distribution.

We are not aware of any mechanism that optimizes the noise addition for the integer summation problem in the aggregator (or general shuffle) model (beyond generic solutions like the geometric mechanism) under the assumption that the input distribution is known.

III. Preliminaries

A. Differential Privacy

DP [3], [4] formalizes the notion of privacy in statistical queries on a data set. Let $X = (x_i)_{0 \le i < n} \in \mathbb{R}^n$ be an input vector of data samples, where each sample x_i belongs to one of n users. Two vectors $X, X' \in \mathbb{R}^n$ are called *adjacent* if their Hamming distance is 1, i.e., if they differ by exactly one element. In this case, we write $X \sim X'$. We denote the set of all possible input vectors by $\mathcal{X} \subset \mathbb{R}^n$. A mechanism M on $\mathcal{X}$ is a (randomized) algorithm that receives $X \in \mathcal{X}$ as an input and outputs into some space $\mathcal{S}$.

Definition 1 (**Approximate Differential Privacy**)**.** A mechanism M is (ε, δ) - approximate differentially private iff

$$\mathsf{Pr}[\mathsf{M}(X) \in S] \le e^{\varepsilon} \cdot \mathsf{Pr}[\mathsf{M}(X') \in S] + \delta \tag{1}$$

holds for all pairs of adjacent data sets $X, X' \in \mathcal{X}$ and all sets of results $S \subset \mathcal{S}$, where the probabilities are in the randomness of M. If $\delta = 0$, then M is ε-differentially private.

Remark 1. If S consists of a single element s (and X, X' are clear from the context) we define $\mathsf{Quot}[\mathsf{M}](s) \coloneqq \frac{\mathsf{Pr}[\mathsf{M}(X)=s]}{\mathsf{Pr}[\mathsf{M}(X')=s]}$ and Eq. (1) becomes for $\delta = 0$ just $\mathsf{Quot}[\mathsf{M}](s) \le e^{\varepsilon}$.

To construct a differentially private mechanism is trivial (and meaningless) without requiring that the mechanism provides a good utility. We measure the utility of a mechanism by how well it approximates a target function g. In this paper, we use the (mean) absolute error as a measure.

Definition 2 (Mean Absolute Error)**.** Let M be a differentially private mechanism and d_X some metric on its output space $\mathcal{S}$ which might depend on the input $X \in \mathcal{X}$. Let g be the target function. We say M approximates g with absolute error $u_{\mathsf{M}}(X)$ at X for

$$u_{\mathsf{M}}(X) = \sum_{s \in \mathcal{S}} \mathsf{Pr}(\mathsf{M}(X) = s) d_X(s, g(X)). \tag{2}$$

We say M (approximates g) with *mean absolute error* (MAE) $u_{\mathsf{M}} = \sum_X \mathsf{Pr}(X) u_{\mathsf{M}}(X)$.

In this paper, we usually choose the taxicab metric on $\mathbb{Z}$ $d_X(\mathsf{M}(X), g(X)) \coloneqq |\mathsf{M}(X) - g(X)|$.

B. (Infinitely) Divisible Distributions

As already described in Section I, a DP-mechanism in the central model that adds additive noise σ can be extended to the multi-party setup if it is divisible.

Definition 3 (Divisibility of Distributions [32])**.** A random variable X is called *decomposable* if there are independent

random variables $X_1, \ldots, X_n$ for some $n \geq 2$ such that $X = X_1 + \cdots + X_n$. It is called *divisible* if the X_i are additionally identically distributed. X is called *infinitely divisible* if for each $n \geq 1$ there is a sequence of i.i.d. random variables $X_1, \ldots, X_n$ with $X = X_1 + \cdots + X_n$.

Remark 2. The binomial distribution $\mathsf{Bin}(p, n)$ for probability p and n trials is divisible, since $\mathsf{Bin}(p, n) = \sum_{i=1}^{n} \mathsf{Bin}(p, 1)$. It is not infinitely divisible; there is, in fact, no infinitely divisible (non-constant) distribution on $\mathbb{Z}$ with finite support.

One of the most commonly used noise distributions in DP is the geometric distribution [13], [33]:

Definition 4 (Symmetric Discrete Geometric Distribution and Mechanism)**.** The (symmetric discrete[2]) geometric distribution on $\mathbb{Z}$ for $\varepsilon > 0$ is defined by the probability density function $\mathsf{Geo}_\varepsilon(x) = \frac{1-e^{-\varepsilon}}{1+e^{-\varepsilon}} e^{-\varepsilon|x|} = \frac{1-q}{1+q} q^{|x|}$ for $q = e^{-\varepsilon}$. The geometric mechanism for an input $x \in \mathbb{Z}$ outputs $x + e$ for $e \leftarrow \mathsf{Geo}_\varepsilon$.

The probability density function of the geometric mechanism makes the connection with DP apparent:

Lemma 1 (DP-privacy of the Geometric Mechanism [33])**.** *The geometric mechanism provides ε-Differential privacy.*

Another interesting property of the geometric distribution is its divisibility:

Lemma 2 (Divisibility of the Geometric Mechanism [18], [34])**.** *The geometric distribution is infinitely divisible: Let $X \sim \mathsf{Geo}_\varepsilon$, then $X = X_1 + \cdots + X_n$ for every $n \geq 1$ and $X_i \sim \mathsf{SGDL}_{1/n}$, where $\mathsf{SGDL}_{1/n}$ denotes the (symmetric generalized discrete) Laplacian distribution [35] for $\beta > 0$ and $q = e^{-\varepsilon}$ on $\mathbb{Z}$ defined by the density*

$$\mathsf{SGDL}_\beta(x) = (1-q)^{2\beta} \sum_{k=|x|}^{\infty} \binom{\beta + k - 1}{k} \binom{\beta + k - |x| - 1}{k - |x|} q^{2k-|x|}.$$

Remark 3. The (symmetric generalized discrete) Laplacian distribution occurs as the difference between two i.i.d. random variables with negative Binomial distribution (with the same parameter). In particular, this relation can be used to sample from the Laplacian distribution.

Also note that (as a consequence) $\mathsf{SGDL}_{\beta_1+\beta_2} = \mathsf{SGDL}_{\beta_1} * \mathsf{SGDL}_{\beta_2}$ holds generally for all $\beta_1, \beta_2 > 0$.

Notation 1. Let χ be a probability distribution on a space $\mathcal{X}$ and f a random variable $\mathcal{X} \to \mathbb{Z}$, then we denote by $f \circ \chi$ the distribution of images of f, i.e. $f \circ \chi(x) = \mathsf{Pr}(f(y) = x | y \leftarrow \chi)$. Similarly, for an algorithm M, $\mathsf{M} \circ \chi$ is the distribution of the outputs over all random coins (from M and χ). If $\mathcal{X} \subset \mathbb{R}^n$, π_j the projection to the j-th component and π_j^c to its complement, then denote $\chi^{w_j}(X) = \mathsf{Pr}(X | \pi_j(X) = w_j)$. Moreover, in slight abuse of notation, we regularly use the same notation for a distribution, its density, or a random variable/DP-mechanism with said (output) distribution. Additionally, we write $\chi^{w_j} \sim_c \chi^{w_j'}$ if $\pi_j^c \circ \chi^{w_j} = \pi_j^c \circ \chi^{w_j'}$.

[2]Since we only work over $\mathbb{Z}$ in this paper, we will usually write just geometric distribution instead of symmetric discrete geometric distribution.

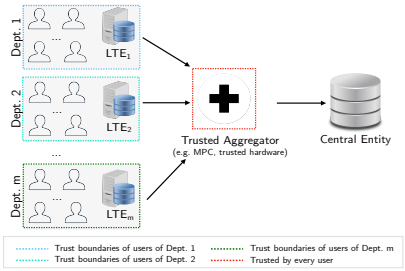

Fig. 3. Setup: each user trusts only their own local LTE and the aggregator.

IV. Setup

In this section, we formalize our setup, explaining the type of adversary that we consider and recalling the trust assumptions that we made in Section I. Our setup (cf. Fig. 3) contains three types of parties: data owners, locally trusted entities (LTEs) and a central entity $\mathcal{C}$.

A. Data Owners

Each data owner (a.k.a. user or employee for our corporate use case) belongs to one locally trusted entity (LTE) and shares all (relevant) private data with this LTE. Data owners will not actively participate in our protocol, but since they provide the sensitive data, we need to protect their privacy.

B. The Central Entity

The central entity $\mathcal{C}$ acts as the analyst in our setup. It can query its subsidiaries, i.e., the LTEs, to provide statistical data. We assume there is a publicly known number m of LTEs. In the integer summation problem discussed here, we assume that each query has the form $(f, f_1, \ldots, f_m)$ for f_i an (integer-valued) *linear* function on the input set of the i-th LTE for $1 \leq i \leq m$ and f linear on $\mathbb{Z}^m$. The central entity wants to compute $f(f_1(X_1), \ldots, f_m(X_m))$, i.e. this is our target function g. We will usually set $f(y_1, \ldots, y_m) \coloneqq \sum_{i=1}^{m} y_i$, i.e., we consider simple aggregation. Other target functions are, however, possible. In particular, averaging (used in federated learning) works completely analogous to aggregation, with the sole difference that the central entity needs to take the average of the result, under the assumption that it knows the number of employees overall or in the different subsidiaries. Note that the central entity is assumed to be actively malicious; it can collude with corrupted parties.

C. Locally Trusted Entities

The LTEs agree on a threshold $t \leq m$ of honest (or even honest-but-curious) LTEs. In particular, these parties are expected not to collude with the central entity. Each locally trusted entity LTE_i collects the raw data $X_i \in \mathcal{X}_i$ from their employees. We assume that each LTE_i estimates the probability that a certain X_i (of its input set $\mathcal{X}_i$) occurs by a distribution χ_i. As already explained in the introduction, the LTE can use previous statistics and iterative Bayesian updates [16] to find an accurate χ_i. These can also be internal statistics that have never been used in the data provided to the central entity. Additionally, it can access all publicly available data,

Input: $X_i, \vec{f} = (f, f_1, \ldots, f_m), \varepsilon, t, \vec{\chi} = (\chi_1, \ldots, \chi_m), \mathcal{X}_i$.
Output: $y_i = f_i(X_1) + e$

1: $D_i(Y_i) \leftarrow \mathsf{Precomputation}(\vec{f}, \varepsilon, \vec{\chi}, \mathcal{X}_i)$ with $D_i(Y_i)$ a probability distribution on $\mathbb{Z}$ for each $Y_i \in \mathcal{X}_i$
2: **return** $y_i = f_i(X_i) + e \leftarrow \mathsf{Sampling}(D_i, X_i, f_i)$

Algorithm 1: Target procedure for LTE_i where Precomputation outputs a family of noise distributions $D_i(Y_i)$ for each $Y_i \in \mathcal{X}_i$, which is independent of the actual data set $X_i \in \mathcal{X}_i$ used to answer the query $\vec{f}$. Sampling outputs an obfuscated LTE_i-internal result $f_i(X_i) + e$ where the noise term e is defined by $D_i(X_i)$.

e.g., the average number of e-mails an employee sends in a mid-size tech company, if f_i asks for the number of e-mails. We assume that the LTEs do not overestimate the obfuscating effect of their input distribution. If in doubt, they should choose a χ_i which obfuscates less. If an LTE underestimates the obfuscating effect of the input distribution, our mechanism (cf. Sections V and VI) adds too much noise and gets less ideal utility, but still better utility than the geometric mechanism (as long as χ_i is not chosen trivially). Naturally, privacy is not affected when adding more noise (cf. Section VI-E for a more detailed discussion).

Upon receiving a query $(f, f_1, \ldots, f_m)$ from the central entity, LTE_i computes f_i on X_i for all $1 \leq i \leq m$. LTE_i runs a randomized algorithm and locally computes some y_i. The process run by LTE_i is described formally in Alg. 1.

In this work, we focus primarily on the first step of selecting the distribution to optimize the mechanism's utility, which, as far as we know, has not yet been explored in the literature.

Once the y_i are computed by all LTE_i, they are entered in the aggregation protocol that outputs $y = f(y_1, \ldots, y_m)$. The aggregation could be facilitated by a trusted third party or a Multi-Party Computation (MPC) protocol. In the latter case, we require the MPC protocol to be actively secure against the corruption of up to $n - t - 1$ LTEs, e.g., [27]. For actively malicious LTEs, this requirement includes secure input protocols like [36] and the protection against replays of other parties' inputs, e.g., through zero-knowledge proofs or commitments. Finally, we note that $\mathcal{C}$ can monitor all communication between the LTEs. It is, therefore, required that the communication is encrypted and leaks no information beyond the output of our mechanism.

Remark 4. In the following, we will set $t = m$, i.e., we consider the case with no actively malicious or colluding LTEs. For the more general case, the LTEs must split the required noise for a certain ε-Differential Privacy (DP) level into t instead of m parts. The sum of the noises of the honest (or honest-but-curious) LTEs under the aggregation protocol will then still be sufficient to reach ε-DP for employees in the honest LTEs. The utility will naturally decrease since, for $t < m$, each party has to add more noise to protect privacy. However, the same holds for the standard Laplacian or Geometric mechanism we compare to. Since we improve accuracy for each LTE separately, the overall utility of our approach cannot be worse than in the aforementioned standard approaches. Of course, unless inputs to the aggregation protocol are checked for consistency, e.g., range checks as used in FL, malicious parties can always enter so much noise that the aggregation result contains (almost) no valuable information about the real statistic anymore.

D. Utility

The utility of a DP mechanism is described in terms of accuracy, i.e., in our setup, how close the output y is to the actual result $g(X_1, \ldots, X_m)$. We measure the accuracy loss in terms of the (mean) absolute error (cf. Definition 2).

V. Single-Party Case

In this section, we want to introduce a new differentially private mechanism for a single LTE_1. We start with the analysis of the state-of-the-art geometric mechanism.

A. Applying the Geometric Mechanism

We have already seen in Section III that the geometric mechanism Geo_ε comes with many favorable properties:

(i) It provides ε-differential privacy (Lemma 1).
(ii) It offers optimal utility in terms of MAE (considering central Differential Privacy (DP) without taking into account the input distribution) [33].
(iii) It is infinitely divisible (Lemma 2) and can, therefore, be used in a multi-party setup, i.e., multiple locally trusted entities (LTEs).

While (iii) will become only relevant once we consider multiple parties in Section VI, it gives a hint of how single-party solutions that are transferable to the multi-party setup could look like.

In this section, we first want to investigate the privacy guarantees of the geometric mechanism given by (i) in more detail. Recall from Definition 4 that the geometric distribution is defined by the density $\mathsf{Geo}_\varepsilon(x) = \frac{1-q}{1+q} q^{|x|}$ for $q = e^{-\varepsilon}$ for $\varepsilon > 0$. Let s be the output of the geometric mechanism and $y = f_1(X), y' = f_1(X')$ two adjacent outputs of f_1, i.e. $|y - y'| = 1$. W.l.o.g. let $y = y' + 1$. Then $\mathsf{Pr}(s|y) = \mathsf{Geo}_\varepsilon(s - y)$ and $\mathsf{Pr}(s|y') = \mathsf{Geo}_\varepsilon(s + 1 - y)$. Thus $\mathsf{Quot}[\mathsf{Geo}_\varepsilon](s) \in \{q, q^{-1}\}$ for all outputs s (cf. Remark 1). Figure 1 shows the behavior of the DP-quotient. In particular, the privacy inequalities (1) in Definition 1 are tightly satisfied for the geometric mechanism applied to the output of f_1, i.e., the standard application of the central model with geometric noise. However, as already described in Section I, our goal is not to protect the result of the internal computation f_1 of LTE_1, but rather the privacy of every single employee, similar to the local model of DP. Let w_j, w_j' be two adjacent secrets of an employee $P_j \in \mathsf{LTE}_1$. We need to ensure that an adversary cannot distinguish whether w_j or w_j' was used to compute an obfuscated result s for any $P_j \in \mathsf{LTE}_1$. If χ_1 is the distribution of inputs $X_1 \in \mathcal{X}_1$ and $\chi_1^{w_j}(X_1)$ the conditional probability to see X_1 if P_j has secret w_j, then we are interested in the

privacy of the (local)[3] mechanism $\mathsf{R}^j_{\mathsf{Geo},\chi_1}$ that upon input w_j samples $Y \leftarrow \chi_1^{w_j}$ and outputs $f_1(X_1)+e, e \leftarrow \mathsf{Geo}_\varepsilon$ for X_1 the $\mathcal{X}_1$-element resulting from w_j and Y. More formally:

Definition 5. Let $\mathsf{M} : \mathcal{X} \subset \mathbb{R}^n \to \mathcal{S}$ be a mechanism, χ a probability distribution on $\mathcal{X}$ and $\mathsf{R}^j_{\mathsf{M},\chi}$ the corresponding local mechanisms[4] for $1 \le j \le n$, i.e., $\mathsf{R}^j_{\mathsf{M},\chi}$ receives the input $w_j \in \{\overline{w}_j \in \mathbb{R} | \exists x \in \mathcal{X} : x_j = \overline{w}_j\}$, samples X from χ^{w_j} and outputs $\mathsf{M}(X)$. Then M is called *ε−differentially private under input distribution* χ if for all $1 \le j \le n$ the $\mathsf{R}^j_{\mathsf{M},\chi}$ are ε-differentially private.

Remark 5. The $\mathsf{R}^j_{\mathsf{M},\chi}$ are ε-locally randomized, and their collection is an LR-oracle in the sense of local differential privacy [6]. However, in local differential privacy, the $\mathsf{R}^j_{\mathsf{M},\chi}$ are usually added up to receive M, while in our approach, the relation to the total mechanism M is much more subtle.

Example 1. Assume an employee takes sick leave (denoted by $w = 1$) to care for sick kids with probability p (independent of the other employees). The holding company is interested in how many people are absent for this reason. E.g., we choose f_1 aggregation and $\chi = \bigtimes_{i=1}^n \mathsf{Bin}(p,1)$ for n the number of employees. Then $f_1(X)$ is distributed as $\mathsf{Bin}(p,n)$ and $\mathsf{Pr}(f_1(X) = s | w = 1) = \mathsf{Bin}(p, n-1)(s-1), \mathsf{Pr}(f_1(X) = s | w = 0) = \mathsf{Bin}(p, n-1)(s)$. The output of the geometric mechanism is then distributed as the convolution $\mathsf{Geo}_\varepsilon * \mathsf{Bin}(p,n)$ of the geometric distribution and a binomial distribution. The DP-quotient is shown in Fig. 1. Observe that the privacy level varies in the obfuscated output s of the mechanism. Namely, on the tails $s \le 0$ and $s \ge n$, the privacy level is still exactly q or q^{-1}, but for more likely values of $f(X)$ and s (cf. Fig. 1 for a similar convolution) the privacy loss is even smaller. These properties hold more generally – see Lemma 3 below, which is proved in Appendix A.

Lemma 3. *Let χ_1 be a distribution on $\mathbb{R}^n$ with support in I_1^n with $I_1 = \{a, \dots, b\}$. Let $w_j, w_j' \in I_1$ with $w_j < w_j'$ and $\chi^{w_j} \sim_c \chi^{w_j'}$. Then $\mathit{Quot}[\mathsf{R}^j_{\mathsf{Geo},\chi}](s) = q^{w_j - w_j'}$ for $s \le a$ and $\mathit{Quot}[\mathsf{R}^j_{\mathsf{Geo},\chi}](s) = q^{w_j' - w_j}$ for $s \ge b$. Furthermore, if $f_1 \circ \chi_1$ has no local minima (outside of the boundaries), then $\mathit{Quot}[\mathsf{R}^j_{\mathsf{Geo},\chi}](s)$ is monotonically decreasing in s.*

We see in Example 1 (and Fig. 1) that the DP-quotient significantly deviates from q (or q^{-1}) for a binomial distribution. This is not surprising since the sum already obfuscates the choice of P_j, e.g., for $n = 50$ employees, a (non-obfuscated) sum 25 can be reached either by $f_1(Y) = 24$ and $w = 1$ or by $f_1(Y) = 25$ and $w = 0$. For some applications, like e-voting, privacy actually only relies on the obfuscation provided by the tally. Namely, it is usually really hard to deduce information about a single voter's choice given the overall tally (see e.g. [37] for a more recent treatment and possible attacks).

For the straightforward application of the geometric mechanism on the output of f_1 we get the ideal utility described in (ii). In particular, we cannot reduce the noise since the privacy inequalities (1) are already tightly satisfied. When we consider $\mathsf{R}^j_{\mathsf{Geo},\chi}$ to measure the privacy loss, we do not change the mechanism employed by LTE_1 – we still only add geometric noise to $f_1(X_1)$. Hence, we still receive the accuracy of the geometric mechanism. However, the geometric mechanism has no longer ideal utility since the privacy loss for a single employee P_j modeled by the DP-quotient for $\mathsf{R}^j_{\mathsf{Geo},\chi}$ is no longer strictly q (or q^{-1}). This allows us to (partly) reduce the noise and improve accuracy. Ultimately, we would like to find a mechanism where the privacy quotient is always in $\{q, q^{-1}\}$ for all $P_j \in \mathsf{LTE}_1$, i.e., an adversary gets the same amount of information independent of the output he sees. Roughly, we want to decrease the noise the mechanism employs whenever $f_1(X_1)$ already obfuscates individual secrets w_j. This reduction of noise will then result in better accuracy.

B. Optimizing the Noise Level for a Single Party

We now propose a new DP mechanism M that uses the obfuscation already employed by the tallying function f_1 to reduce the noise level and, thereby, improve the mechanism's accuracy while maintaining the same privacy level as the geometric mechanism. As mentioned above in Section V-A, the geometric mechanism Geo_ε, which adds geometric noise to $f_1(X_1)$, already has good accuracy. We can compute the mean absolute error (MAE) of the geometric mechanism explicitly (cf. Definition 2): $u_{\mathsf{Geo}_\varepsilon}(X) = \sum_{s\in\mathbb{Z}} \mathsf{Geo}_\varepsilon(s)|s| = \frac{1-q}{1+q}\sum_{s\in\mathbb{Z}} q^{|s|}|s| = \frac{1-q}{1+q}\sum_{s\in\mathbb{N}} 2\sum_{k=1}^{s} q^k = \frac{1-q}{1+q}\cdot\frac{2q}{(1-q)^2} = \frac{2q}{1-q^2}$ independent of X. Hence $u_{\mathsf{Geo}_\varepsilon} = u_{\mathsf{Geo}_\varepsilon}(X)$.

Since we want to optimize the geometric mechanism in our setup, our new mechanism M should satisfy $u_{\mathsf{M}}(X) \le \frac{2q}{1+q^2}$ for all X. In particular, the MAE of M can then not be larger than $u_{\mathsf{Geo}_\varepsilon}$. Recall that we assume that $f_1 : \mathbb{R}^n \supset \mathcal{X}_1 \to \mathbb{Z}$ is linear, e.g., simple aggregation of the components. Furthermore, $f_1 \circ \chi_1^w \sim_c f_1 \circ \chi_1^{w'}, \forall x \in \mathbb{Z}$, i.e., the distribution of the sum of all other employees is independent of the value of w or w', respectively. To simplify notation we represent M by an infinite probability matrix $(r_{x,s})_{x,s\in\mathbb{Z}}$ such that $r_{f(X_1),s} = \mathsf{Pr}(\mathsf{M}(X_1) = s)$. Note that $\sum_{s\in\mathbb{Z}} r_{x,s} = 1$ must hold for all $x \in \mathbb{Z}$ for a probability matrix. The absolute error condition in this notation reads

$$\sum_{s\in\mathbb{Z}} r_{x,s}|x-s| \le \frac{2q}{1-q^2}, \quad x \in \mathbb{Z}. \tag{3}$$

To get ε-DP under input distribution χ_1 (cf. Def. 5), consider adjacent inputs w_j, w_j'. Since f_1 is chosen linear, we get

$$\begin{aligned}\mathsf{Pr}(\mathsf{R}^j_{\mathsf{M},\chi_1}(w_j) = s) &= \sum_{x\in\mathbb{Z}} f_1 \circ \chi_1^{w_j}(x) r_{x,s} \\ &= \sum_{x\in\mathbb{Z}} f_1 \circ \chi_1^{0}(x) r_{x+f_1(w_j,0),s}.\end{aligned}$$

If we apply the same calculation to w', we can deduce the following privacy conditions:

$$q \cdot \mathsf{Pr}(\mathsf{R}^j_{\mathsf{M},\chi_1}(w_j) = s) = q \sum_{x\in\mathbb{Z}} f_1 \circ \chi_1^{0}(x) r_{x+f_1(w_j,0),s}$$

[3] *Local* is used in reference to the local model of DP – see Remark 5.

[4] To simplify notation, we occasionally drop the j, M and/or χ indices if they are clear from context.

$$\leq \sum_{x\in\mathbb{Z}} f_1 \circ \chi_1^0(x) r_{x+f_1(w'_j,0),s} = \mathsf{Pr}(\mathsf{R}^j_{\mathsf{M},\chi_1}(w'_j) = s). \quad (4)$$

Remark 6. A priori, we need to check Eq. (4) for all possible secrets w_j, w'_j of the P_j to guarantee ε-DP. However, since$\mathsf{Pr}(\mathsf{R}^j_{\mathsf{M},\chi_1}(w_j) = s)/\mathsf{Pr}(\mathsf{R}^j_{\mathsf{M},\chi_1}(w_j+1) = s) \cdot \mathsf{Pr}(\mathsf{R}^j_{\mathsf{M},\chi_1}(w_j+1) = s)/\mathsf{Pr}(\mathsf{R}^j_{\mathsf{M},\chi_1}(w_j+2) = s)$ is equal to $\mathsf{Pr}(\mathsf{R}^j_{\mathsf{M},\chi_1}(w_j) = s)/\mathsf{Pr}(\mathsf{R}^j_{\mathsf{M},\chi_1}(w_j+2) = s)$, it is enough to consider the case $|w_j - w'_j| = 1$. If we can prove the DP-quotient smaller e^ε for these cases, then the overall privacy level will be bounded by $\mathsf{sens} \cdot \varepsilon$, where $\mathsf{sens} = \max_{P_j\in\mathsf{LTE}_1,w_j,w'_j} |w_j - w'_j|$ is the sensitivity.

Note that all conditions, i.e., the conditions Eqs. (3) and (4) and conditions on a probability matrix like $\sum_{s\in\mathbb{Z}} r_{x,s}$ are linear. If we assume that $r_{x,s} = 0$ whenever $x \notin I_1 := \{a_1, \dots, b_1\}$ or $s \notin I_1$ for some $a_1, b_1 \in \mathbb{Z}$, we can minimize a linear function like the mean absolute error

$$u_\mathsf{M} = \sum_{x,s\in\mathbb{Z}} f_1 \circ \chi_1(x) r_{x,s} |s - x|$$

efficiently with generic methods like the interior point method and linear programming, e.g., [38]. The assumption $r_{x,s} = 0$ for $x \notin I_1$ is usually satisfied in real-world applications since the input space is generally bounded, and then f_1 has a bounded image. The assumption $r_{x,s} = 0$ for $s \notin I_1$ is generally not satisfied, e.g., not satisfied by the geometric mechanism. Even worse, as we have seen in Remark 2, no finitely supported distribution is infinitely divisible (apart from point distributions).[5]

Fortunately, in the single-party case, we can compute an infinitely supported distribution efficiently. As we have seen in Fig. 1 and Lemma 3, the geometric distribution satisfies the privacy conditions already tightly on the tails. We therefore set $r_{x,s} = \mathsf{Geo}_\varepsilon(x-s)$ for all $x \in I_1$ and $s \notin I_1$ for a, b such that $a_1 \leq f_1 \leq b_1$ uniformly. The additional weight outside of I_1 reduces the weight available inside the interval. Namely, we have to replace the condition $\sum_{s\in\mathbb{Z}} r_{x,s} = 1$ by $\sum_{s\in I_1} r_{x,s} = 1 - \sum_{s\notin I_1} \mathsf{Geo}_\varepsilon(x-s) = 1 - (q^{x-b} + q^{a-x}) \cdot \frac{q}{1+q} =: c$.

The conditions in Eqs. (3) and (4) stay untouched, and we can again use a generic algorithm like [38] to find a distribution $(r_{x,s})_{x,s\in\mathbb{Z}}$ with minimal MAE efficiently. In particular, a solution must exist since the trivial continuation $r_{x,s} = \mathsf{Geo}(x-s), \forall x, s \in \mathbb{Z}$, obviously satisfies all conditions. Overall, we get the following result.

Theorem 1. *The mechanism* M *with output distribution* $\mathsf{Pr}(\mathsf{M}(X_1) = s) = r_{f_1(X_1),s}, \forall X_1 \in \mathcal{X}_1$ *approximates* f_1 *with minimal MAE under all mechanisms with (infinitely supported) output distribution that satisfies Eqs.* (3) *and* (4)*. The family of output distributions* $(r_{f(X_1),s})_{s\in\mathbb{Z}}$ *can be computed in polynomial time (in* $|f_1(\mathcal{X}_1)|$*).*

[5]We remark that truncated versions of the geometric mechanism and our single-party mechanism exist. However, they are not divisible and do not generalize to the multi-party setup discussed later in Section VI.

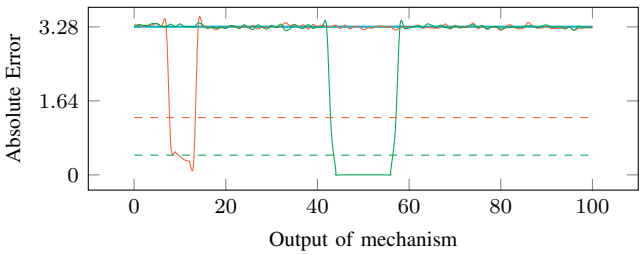

Fig. 4. Absolute error u_M of our mechanism M (cf. Section V-B) with input distribution $\chi = \mathsf{Bin}(p, 100)$ and Geo_ε (blue at 3.28) for $p = 0.1$ (solid red) and $p = 0.5$ (solid green). The dashed lines show the corresponding mean absolute error for $p = 0.1$ (red) and $p = 0.5$ (green).

Example 2. In special cases, where χ_1 is nicely shaped, e.g., similar to the binomial distribution, we can also directly construct a simple algorithm like Alg. 2 to find a near minimal solution for the MAE.

In Alg. 2 we focus on those outputs x of f_1 which improve privacy in Eq. (4), i.e. for x with $\omega^{f_1\circ\chi_1}_{q,1,w_j}(x) := qf_1 \circ \chi_1^0(x - f_1(w_j, 0)) - f_1 \circ \chi_1^0(x - f_1(w'_j, 0)) \leq 0$ and $\omega^{f_1\circ\chi_1}_{q^{-1},1,w_j}(x) \geq 0$. Note that here we already use Remark 6 and only consider $w'_j = w_j \pm 1$.

For $f_1 \circ \chi_1^0$ a binomial distribution (cf. our use-case example in Section I), such x exists since (for sufficiently large n) around the mode, two adjacent probabilities will not differ by more than a factor q, i.e., the curve becomes flat enough. The same holds for distributions without a local minimum, i.e., those that increase monotonically to maximum and then fall off monotonically. We then find a set $V = \{v_0, \dots, v_1\}$ where both inequalities hold. In fact for $f \circ \chi^0 = \mathsf{Bin}_{p,n-1}$, we can explicitly compute $v_0 := \frac{qpn}{qp+1-p} \leq x$ and $v_1 := \frac{pn}{p+q(1-p)}$. Inside of V, we can increase diagonal values $r_{x,x}$, i.e., output the actual $s = x$, which improves accuracy (and privacy inside of V). Of course, to receive a probability distribution, we then have to lower other $r_{x,s'}$, which will decrease privacy in other columns s'. Fortunately, we know from Lemma 3 that inside of $\{0, \dots, n\}$ the privacy condition in Eq. (4) is no longer tight (cf. Fig. 1), which means that there is room for improvement. Thus, we can decrease some of the $r_{x,s'}$ without violating ε-privacy. Hence, this weight shifting in the direction of the actual result $s = x$ is possible and improves our MAE.

Figure 4 shows generic benchmarks of Alg. 2 against the geometric mechanism. Our approach comes with a significantly smaller error compared to the geometric mechanism (both for the same privacy level). In particular, the MAE converges to 0 for large n. The optimization interval V is clearly recognizable; V gets smaller for more extreme p.

While smaller improvements to Alg. 2 are possible (Remark 7), Alg. 2 approximates the minimal MAE well for large values of n. Namely, note that $v_1 - v_0 = \frac{pn(q^{-1}-q)}{p+q(1-p)}$ is linear in n and hence $\sum_{x\in\{v_0,\dots,v_1\}} f \circ \chi(x) \xrightarrow{n\to\infty} 1$. In particular, for large n, the MAE is dominated by the values in $\{v_0, \dots, v_1\}$. Hence, our algorithm outputs an MAE, which converges against the minimal possible MAE for $n \to \infty$. In Remark 7, we show

that further optimizations for smaller n are possible.

Remark 7. The probability distribution $(r_{x,s})_{x,s\in\mathbb{Z}}$ reached by Alg. 2 is not ideal. For example, if $\omega_{q,1,w_j}^{f_1\circ\chi_1}(x), \omega_{q,1,w_j}^{f_1\circ\chi_1}(y) > 0$, i.e. x, y not in V then we can set $r_{x,x} \leftarrow r_{x,x} + \delta, r_{x,y} \leftarrow r_{x,y} - \delta, r_{y,y} \leftarrow r_{y,y} + \delta', r_{y,x} \leftarrow r_{x,y} - \delta'$ for $\delta \cdot \omega_{q,1,w_j}^{f_1\circ\chi_1}(x) = \delta' \cdot \omega_{q,1,w_j}^{f_1\circ\chi_1}(y) > 0$. In column x, the positive shift by $r_{x,x}+\delta$ in (4) is compensated by $r_{y,x} - \delta'$ and hence privacy is preserved under this shift. Analogously for column y. However, the diagonal entries $r_{x,x}, r_{y,y}$ are both increased, which improves accuracy. However, the effect of these optimizations is negligible for large n since the cases outside of V occur with low probability. We omit these optimizations in Alg. 2.

Remark 8. We want to briefly discuss how LTE_1 can sample according to $(r_{x,s})_{x,s\in\mathbb{Z}}$, i.e., how our mechanism M is realized. LTE_1 first samples a geometric noise $e \in \mathbb{Z}$, e.g. by taking the difference of two i.i.d. random variables with negative binomial distribution. (i) If $s = x + e \notin I_1$ output $s = x + e$; (ii) else sample according to the finite distribution $(c^{-1} r_{x,s})_{x,s\in I_1}$. If Alg. 2 was used to generate $(r_{x,s})_{x,s\in I_1}$, LTE_1 could also replace (ii) by (ii-1) else, if $x = s$ output s; (ii-2) else, sample a bit b with probability $r_{x,s}/\mathsf{Geo}(s-x)$ and output s if $b = 1$, and x otherwise.

VI. Multi- party Case

In this section, we turn to the multi-party setup. Namely, we have multiple LTEs that need to jointly compute a statistic over distributed private data. As described in Sections I and IV and explicitly in Alg. 1, each LTE_i receives a target function f_i to be evaluated on their secret data set X_i. LTE_i runs a local mechanism M_i to generate a noise distribution, which is used to obfuscate the local result $f_i(X_i)$.

In the previous Section V we have seen how we can reduce the added noise in the single-party case whenever $f_1(X_1)$ already (partly) obfuscates the individual samples. In the multi-party case, the situation is more complex since an LTE, say LTE_1, no longer knows the actual output $f(f_1(X_1), \ldots, f_m(X_1))$ that the central entity queried. For example, let all f_i and f be aggregations. Now, if $f_1(X_1)$ is close to the mode of $f_i \circ \chi_i$ then in the single-party case, we could reduce the noise (cf. Fig. 4). However, the other $f_j(X_j)$ for $j > 1$ could all be extremely small, and thus $f(f_1(X_1), \ldots, f_m(X_m))$ can be far away from the mode of the total distribution $f \circ (f_1 \circ \chi_1, \ldots, f_m \circ \chi_m)$. In particular, the LTEs should generate a large amount of noise (similar to the geometric noise).

Technically, this problem becomes visible when we consider the convolution of the different noise distributions. Recall from Section III that the geometric noise decomposes nicely into (symmetric generalized discrete) Laplacian noise $\mathsf{SGDL}_{1/m}$ by the infinite divisibility property of Geo_ε. Now, if we apply our solution from the single-party case, we deviate from $\mathsf{SGDL}_{1/m}$ whenever the $f_1(X_1)$ is close to its mode, i.e., consider $r_{x,s}$ as in Section V where the tails remain $\mathsf{SGDL}_{1/m}(x-s)$. Unfortunately, this deviation is spread over all of $\mathbb{Z}$ under convolution. In particular, the local modification in one LTE lets the convolution deviate from the geometric distribution on the tails, too. But the geometric distribution was ideal on the tails, such that the privacy relations were already tightly satisfied (cf. Fig. 1). One can show that a modification as in Section V increases the DP-quotient on the tails and hence breaks ε-DP (cf. Remark 9 in Appendix A) .

We, therefore, need to refine the technique used in the single-party case. In our new mechanism, all LTEs run the same mechanism, although with different inputs. Hence, we assume w.l.o.g. that our individual data to be protected is in LTE_1. Let χ_i be the distributions of secrets of LTE_i. Denote by η_i the distribution of the noisy output of LTE_i after applying our mechanism, i.e. the distribution of $\mathsf{M}_i \circ f_i \circ \chi_i$. As usual, we assume that the mechanism M_i is public. Since the χ_i, f_i (cf. Alg. 1) are also available, we assume for now that all LTEs know all η_i.[6] Let $\eta^\neg = \eta_2 * \cdots * \eta_m$ be the noisy distribution of the sum of all LTE_i without LTE_1. Let $\eta = \eta_1 * \eta^\neg$. As before we denote by $r_{f_1(X),s} = \mathsf{Pr}(\mathsf{M}_1(X) = s)$ our probability matrix. Furthermore, let $I_1 = \{a_1, \ldots, b_1\} \supset f(f_1(\mathcal{X}_1), 0)$ and $I = \{a, \ldots, b\} \supset f(f_1(\mathcal{X}_1), \ldots, f_m(\mathcal{X}_m))$.

In the multi-party setup, the privacy conditions (4) become

$$\sum_{s_1\in\mathbb{Z}} \eta^\neg(s - s_1) \sum_{x\in I_1} \omega_{q,1,w_j}^{f_1\circ\chi_1}(x) r_{x,s_1} \le 0 \tag{5}$$

$$\sum_{s_1\in\mathbb{Z}} \eta^\neg(s - s_1) \sum_{x\in I_1} \omega_{q^{-1},1,w_j}^{f_1\circ\chi_1}(x) r_{x,s_1} \ge 0 \tag{6}$$

for w_j a secret of $P_j \in \mathsf{LTE}_1$ and $\omega_{q,1,w_j}^{\mu}(x) = qf \circ \mu^0(x - f(f_1(w_j, 0), 0)) - f \circ \mu^0(x - f(f_1(w_j + 1, 0), 0))$ for any distribution μ on $\mathbb{Z}^m$ – we use $w_j' = w_j + 1$ as described in Remark 6. Note that $\omega_{q,1,w_j}^{\mu}$ does not depend on w_j, but just on $w_j - w_j' = 1$, since we always compare a distribution shifted by $f \circ f_1(w_j)$ with one shifted by $f \circ f_1(w_j')$.

Note that if LTE_1 changes $r_{x,s}$, this will affect η_1 and analogously for all other LTEs. In particular, if LTE_1 changes $r_{x,s}$, the respective condition in (5) and (6) for LTE_2 change. If LTE_2 adapts their distribution to this modification, this will result in a different $\eta^\neg$, and the initial solution of LTE_1 might no longer be differentially private. To avoid these mutual dependencies and to allow each LTE to optimize its noise distribution independently, we aim for $r_{x,s}$ that preserves a previously fixed η_1. Therefore each LTE_i chooses a new distribution μ_i on $\mathbb{Z}$ and sets $\eta_i = \mu_i * \mathsf{SGDL}_{1/m}$. We then impose a new condition

$$\sum_{x\in\mathbb{Z}} f_1 \circ \chi_1(x) r_{x,s} = \eta_1(s), \forall s \in \mathbb{Z} \tag{7}$$

for LTE_1 and similar for all other LTE_i. In particular, $\eta^\neg = \mu^\neg * \mathsf{SGDL}_{m-1/m}$ for $\mu^\neg := \mu_2 * \cdots * \mu_m$ and $\eta = \mu_1 * \cdots * \mu_m * \mathsf{Geo}_\varepsilon$. Let's briefly discuss the motivation behind μ_1. In Fig. 2, we see that $\chi_1 * \mathsf{SGDL}_{1/m}$, i.e., the distribution that results from the standard Laplacian noise, smoothes out χ_1, which is the standard behavior under convolution. As a result, $\chi_1 * \mathsf{SGDL}_{1/m}$ and χ_1 differ most around the mode, i.e., the most

[6] For our mechanism in Section VI-C the LTEs do not require exact knowledge of η_i (cf. Section VI-D).

likely value. If we choose a μ_1 more concentrated than χ_1, $\mu_1 * \mathsf{SGDL}_{1/m}$ matches χ_1 better. While closer probabilities or, as a result, a smaller statistical distance of the output distributions of LTE_1 does not automatically result in a lower MAE, it makes our condition in Eq. (7) less tight close to the mode. As in the single-party case, the values close to the mode are the ones that are already most obfuscated by $f_1(X_1)$ and offer the biggest chance of noise reduction. With a larger value of $\eta_1(s)$ in Eq. (7) close to the mode, we can later assign more weight to actual results $f_1(X_1)$ around the mode.

A. Accuracy on the Tails

We want to discuss next what requirements μ_1 has to satisfy to be used in our mechanism. Similar to the single-party case, we cannot expect to outperform the standard Laplacian/geometric mechanism on the tails since they already lead to tight DP-conditions there. In order to satisfy Eq. (7) outside of some interval I_1, i.e. on the tails, we proceed as follows: LTE_1 runs an algorithm A on the result $f_1(X_1)$ it receives. As before, denote by $A_{x,y} = \mathsf{Pr}(A(x) = y)$ where we require $y \in I_1$, too. Define $\mu_1 = A \circ \chi_1$. The LTE then samples Laplacian noise $e \leftarrow \mathsf{SGDL}_{1/m}$ and outputs $y + e$, whenever $s = y + e \notin I_1$. By construction, $\mathsf{Pr}(s \notin I_1) = \sum_{x,y \in I_1} f_1 \circ \chi_1(x) A_{x,y} \mathsf{SGDL}(s-y) = \mu_1 * \mathsf{SGDL}(s) = \eta_1(s)$. Hence, if we find a suitable A, Eq. (7) holds outside a finite set, and we can hope to get to a finite set of linear (in)equalities, we can optimize similar to Section V.

Obviously, the output of a non-trivial A is less accurate than the actual result $f_1(X_1)$. If we add $\mathsf{SGDL}_{1/m}$-noise to both, the latter will still be more accurate. Fortunately, the tails of the new distribution are not negatively affected and contribute as much to the absolute error as in the standard mechanism. Namely, if we apply a $\mathsf{SGDL}_{1/m}$-noise directly to x then we get an error e with probability $2 \cdot \mathsf{SGDL}(e)$, i.e. the contribution to the absolute error is $\sum_{e \in \mathbb{N}} 2 \cdot \mathsf{SGDL}(e)|e|$. If instead we apply A and then $\mathsf{SGDL}_{1/m}$, the absolute error for x becomes $\sum_{e \in \mathbb{Z}} \mathsf{SGDL}(e) \sum_{y \in I_1} A_{x,y} |y + e - x|$. Now consider only noise $e \geq E := \max\{b - x, x - a\}$. Then we have $\sum_{e \geq E} \mathsf{SGDL}(e) \sum_{y \in I_1} A_{x,y} (|y + e - x| + |y - e - x|) = \sum_{e \geq E} \mathsf{SGDL}(e) \sum_{y \in I_1} A_{x,y} (y + e - x + e + x - y) = 2 \sum_{e \geq E} \mathsf{SGDL}(e)|e|$. Thus for $e \geq E$ the contribution to the absolute error remains the same. Hence, we can formulate an MAE condition, which only depends on a finite interval:

$$\sum_{e < |E|} \sum_{y \in I_1} A_{x,y} |y + e - x| \cdot r_{x,y+e} \leq 2 \sum_{e=1}^{E-1} \mathsf{SGDL}(e)|e|. \quad (8)$$

If Eq. (8) holds, then the absolute error of our new mechanism for each x is bounded from above by the error of the Laplacian mechanism.

B. Privacy on the Tails

Next, we turn to privacy and again consider the tails first.

Claim 1. For $s \leq a$ we can rewrite (5) as[7]

$$\sum_{s_1 \in I_1} \eta^{\neg}(s - s_1) u_q(s_1) \leq 0 \quad (9)$$

$$u_q(s_1) := -\omega^{\mu_1}_{q,1,w_j} * \mathsf{SGDL}_{1/m}(s_1) + \sum_{x \in I_1} \omega^{f_1 \circ \chi_1}_{q,1,w_j}(x) r_{x,s_1}$$

Proof. First note that the convolution with the first term of u_q is 0, since the $\mathsf{SGDL}_{1/m}$-part of u_q convolutes with the $\mathsf{SGDL}_{m-1/m}$-part of $\eta^{\neg}$ to a geometric distribution. But we know from Lemma 3 that for any distribution μ with support in I, we have $q \cdot \mu * \mathsf{Geo}(s) = \mu * \mathsf{Geo}(s-1)$ for $s \leq a$, i.e. $\omega^{\mu_1}_{q,1,w_j} * \mu_2 \cdots * \mu_m * \mathsf{Geo}_{\varepsilon} = 0$. Moreover, recall from the last paragraph that $r_{x,s_1} = \sum_{y \in I_1} A_{x,y} \mathsf{SGDL}_{1/m}(s_1 - y)$ for $s_1 \notin I_1$ and hence $\sum_{x \in I_1} f_1 \circ \chi_1(x) r_{x,s_1} = \mu_1 * \mathsf{SGDL}_{1/m}(s_1)$ for $s_1 \notin I_1$. In particular, for $s_1 \notin I_1$, the two terms of u_q cancel out, and we can restrict the summation to $s_1 \in I_1$ in Eq. (9). □

Claim 2. For $s \geq b$ Eq. (5) is equivalent to

$$(\mu^{\neg} * u_q) * \mathsf{SGDL}_{m-1/m}(s) \leq (q^{-1} - q) \cdot \mu * \mathsf{Geo}(s). \quad (10)$$

Proof. The first term of u_q is no longer convolved to 0 for $s \geq a$ since the privacy quotient is close to q, not q^{-1}. Hence Eq. (5) becomes equivalent to

$$\begin{aligned} &\sum_{s_1 \in I_1} \eta^{\neg}(s - s_1) u_q(s_1) \\ &\leq - \sum_{s_1 \in I_1} \eta^{\neg}(s - s_1) \cdot \omega^{\mu_1}_{q,1,w_j} * \mathsf{SGDL}_{1/m}(s_1) \\ &= - \sum_{x \in I_1} \omega^{\mu_1}_{q,1,w_j}(x) (\mu^{\neg} * \mathsf{Geo})(s - x). \end{aligned} \quad (11)$$

Since $s - x$ is not in the support of $\mu^{\neg}$ by definition of I_1, I and linearity of f, we can again use that the DP-quotient for $s \geq b$ becomes tight, i.e.$(\mu^{\neg} * \mathsf{Geo})(s + 1 - x_0) = q(\mu^{\neg} * \mathsf{Geo})(s - x_0)$ for all $x_0 \in I_1, \forall s \geq b$. Hence in this case $-\sum_{x \in I_1} \omega^{\mu_1}_{q,1,w_j}(x) (\mu^{\neg} * \mathsf{Geo})(s - x) = \sum_{x \in I_1} (q^{-1} - q) \mu_1(x) (\mu^{\neg} * \mathsf{Geo})(s - x) = (q^{-1} - q)(\mu * \mathsf{Geo})(s)$ for $\mu = \mu_1 * \mu^{\neg}$. This leads to Eq. (10). □

Thus, we transformed Eq. (5) into condition (9) for $s \leq a$, condition (10) for $s \geq b$ and a finite number of conditions (11) for (the remaining cases of) $s \in I$.

Analogously to (5) in Claim 1 and 2 we can handle (6) by exchanging q with q^{-1}, $\leq$ with $\geq$ and a with b. Moreover, we only need a finite number of equations to ensure Eq. (7) as discussed. Furthermore, we have also seen that we only need a finite number of conditions in Eq. (8) to ensure that the absolute errors decrease. Finally, there are obviously only a finite number of conditions to ensure that $(r_{x,s})_{x \in I_1, s \in I}$ becomes a probability matrix. The optimization of the overall MAE is again, as in Section V, a problem from linear programming. Hence, if we can ensure that at least one solution exists, then we can find the ideal solution efficiently, e.g., using [39].

[7] Recall that $I = \{a, \ldots, b\}$ contains the image of $f(f_1, \ldots, f_m)$.

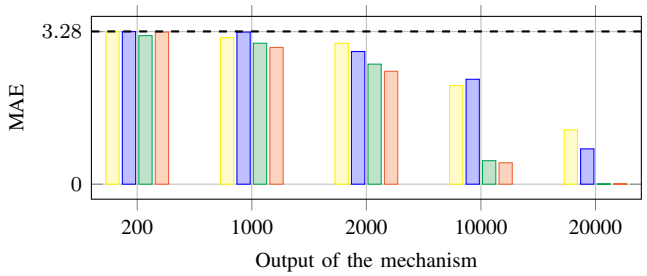

Fig. 5. MAE with our mechanism for m parties for $m = 2$ (green, blue), $m = 5$ (yellow, black); binomially distributed $f_i \circ \chi_i = \mathsf{Bin}(p_i, n_i)$ with $p_i = 0.i$ (green, red) and $p_i = 0.05$ (blue, yellow) and $n_i = n/m$ for LTE_i. Moreover, Geo_ε-MAE (black dashed) is the baseline.

C. Shifting Weights Given Boundaries Conditions

Similar to Section V, we can also construct a mechanism explicitly, i.e., without relying on generic optimization methods from linear programming.

To find a mechanism with better absolute error than the Laplacian mechanism, we start with the trivially private solution $r_{x,s} = \sum_{y \in I_1} A_{x,y} \mathsf{SGDL}_{1/m}(s-y)$ for all $x \in I_1, s \in I$. Unless $\mu_1 = f_1 \circ \chi_1$, this solution will not improve the utility, though, since A increases the error. However, all other conditions hold. We now want to shift weights between the different $r_{x,s}$ such that none of the constraints is violated and such that the absolute error decreases.

We modify r_{x,s_1} by $\Delta(x) \in \mathbb{R}$. To keep the row sum constant (i.e., to receive a probability distribution), we change $r_{x,s_2} \leftarrow r_{x,s_2} - \Delta(x)$ for some s_2. To keep the column sum (i.e., keep the output distribution $\mu_1 * \mathsf{SGDL}_{1/m}$), this implies that $\sum_{x \in I_1} f \circ f_1 \circ \chi_1(x)\Delta(x) = 0$. Furthermore $0 \leq r_{x,s_1} + \Delta(x) \leq 1$ and $0 \leq r_{x,s_2} - \Delta(x) \leq 1$ to get probabilities. The change also affects u_q and $u_{q^{-1}}$. Namely, $u_q(s_1)$ $+=$ $\sum_{x \in I_1} \omega_{q,1,w_j}^{f_1 \circ \chi_1}(x)\Delta(x), u_q(s_2)$ $-=$ $\sum_{x \in I_1} \omega_{q,1,w_j}^{f_1 \circ \chi_1}(x)\Delta(x)$. Our boundary condition (9) then implies that

$$\begin{aligned} &\eta^\neg(s-s_1)u_q(s_1) + \eta^\neg(s-s_2)u_q(s_2) \leq 0 \\ &\Leftrightarrow (\eta^\neg(s-s_1) - \eta^\neg(s-s_2)) \sum_{x \in I_1} \omega_{q,1,w_j}^{f_1 \circ \chi_1}(x)\Delta(x) \leq 0. \end{aligned}$$

We first consider the case where $\eta^\neg$ increases, i.e. $\eta^\neg(s-s_1) > \eta^\neg(s-s_2)$ for $s_1 < s_2$. Then we get the privacy condition

$$\sum_{x \in I_1} \omega_{q,1,w_j}^{f_1 \circ \chi_1}(x)\Delta(x) \leq 0. \tag{12}$$

We get analogously that for $\eta^\neg$ decreasing (and $s_1 < s_2$) the $\omega_{q^{-1},1,w_j}^{f_1 \circ \chi_1}$ condition is equivalent to $\sum_{x \in I_1} \omega_{q^{-1},1,w_j}^{f_1 \circ \chi_1}(x)\Delta(x) \leq 0$. Note that for $s \leq a$, $\eta^\neg$ will increase (since it is a convolution with a Laplacian/geometric distribution) and decrease for $s \geq b$. The remaining equations Eq. (10) can be treated analogously. Namely, for $s_1 < s_2, \eta^\neg$ decreasing Eq. (10) becomes $\sum_{x \in I_1} \omega_{q,1,w_j}^{f_1 \circ \chi_1}(x)\Delta(x) \geq {(q^{-1}-q) \cdot \mu * \mathsf{Geo}(s)}/{(\eta^\neg(s-s_1) - \eta^\neg(s-s_2))}$. Since the Laplacian fall-off is dominating the geometric one (for $\beta = {m-1}/{m}$), this condition holds for $b \geq s$ whenever it holds for $b = s$. A similar statement holds for $\eta^\neg$ increasing and $s \leq a$. Overall, we get a finite number of privacy conditions and could now even employ standard techniques for finitely constrained linear programs, e.g. [38]. For our optimization, we can, however, proceed without these generic methods.

While Eq. (12) is an inequality, we restrict to optimizations that tightly satisfy Eq. (12) in the following.[8] Note that solutions with strictly negative terms would improve privacy. However, our goal is to make the privacy conditions (1) tighter. But this is already done by using a more concentrated function μ_1. We do not want to undo this achievement.[9] We remark that as a positive side effect, Eqs. (10) and (11) also hold.

We have now found two optimization problems: depending on whether $\eta^\neg(s-s_1) - \eta^\neg(s-s_2)$ is increasing or decreasing (if we choose $s_1 < s_2$). For our standard example of distributions with at most one local maximum, this means that the optimization only depends on the mode. We use this fact below in Section VI-D. We can solve the linear system of equations and then systematically shift weights until a minimal MAE is reached. We discuss this technique in detail in Appendix B.

There are natural limits to our shifting technique. Namely, probabilities cannot fall below 0, i.e., we can no longer take weights of one row, say x, to improve the MAE for another x'. At some point, all solutions Δ to our system of linear equations have $r_{x,s_1} = 0$ and $\Delta(x) \leq 0$, i.e., no shift is possible. Then, our optimization terminates. Furthermore, the stronger A_1 and μ_1 deviate from $f_1 \circ \chi_1$, the worse the absolute error of our trivial starting solution, and the optimization might never even reach the MAE of the geometric mechanism. However, the more concentrated μ_1, the less weight is put by $\mu_1 * \mathsf{SGDL}_{1/m}$ on the tails, and the more weight is available to more accurately output the actual values (cf. Fig. 2). It is, therefore, important to choose μ_1 correctly.

D. Choice of an approximating distribution μ

Finally, we turn to choosing a suitable approximation μ_1. By construction of our optimization algorithm above, each LTE_i needs to preserve the mode of its own $f_i \circ \chi_i$ (or, more generally, all minima and maxima). For the general approach to scaling χ while preserving the maximum, we refer to scaling theory, e.g., in [40] and follow-up papers. We note that in special cases, e.g. χ_i binomially distributed, even the dependence on the mode drops, since the other constraints ensure that all privacy conditions hold once Eq. (12) holds (cf. Appendix B). Then each LTE can apply our mechanism without any knowledge of the other LTEs' input distributions.

In our special case of a distribution with one maximum, we simply test different shifts for A and check whether they lead to an optimization. Namely, we start with the least likely output x_1 of f_1. We then set $A_{x_1,x_1} = 1-\alpha$ and $A_{x_1,x_1+1} = \alpha$ for a small shift α and $x_1 + 1$ closer to the mode than x_1. If $x_1 - 1$ is closer to the mode, then we instead set $A_{x_1,x_1-1} = \alpha$.

[8] Moreover, recall that the Eq. (12) is independent of w_j, i.e., we only have one equation.

[9] We also ran optimizations where we allowed strictly negative values in Eq. (12). As expected, the optimized results still tightly satisfied Eq. (12). For a general discuss of optimality and tightness see also Remark 10.

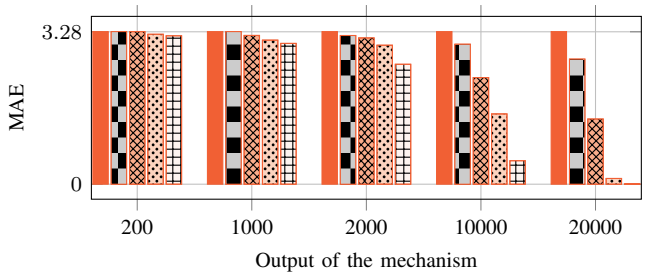

Fig. 6. The right-most bar (red in Fig. 5) is the MAE with our mechanism for 2 parties with probabilities $\mathsf{Bin}(p_i, n_i)$ for $p_1 = 0.1, p_2 = 0.2, n_1 = n_2 = n/2$. The bars to the left describe lower variance prior distributions $f_i \circ \chi_i$ with variance descending by factors $0.6, 0.3, 0.1$ (all with the same mean of $p_i n_i$). The left-most bar describes a δ-function around the same mean.

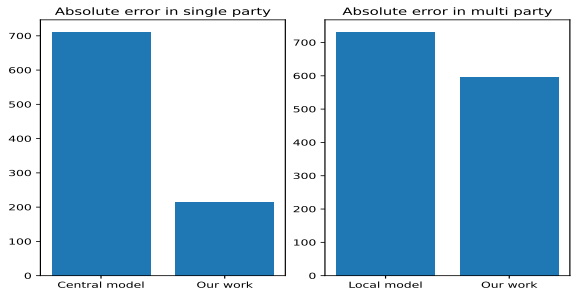

Fig. 7. Comparison between the central and local models with our methods.

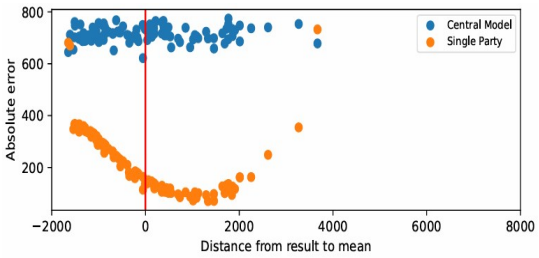

Fig. 8. Absolute error with respect to the distance from the expected value.

Now, we run our optimization, and if the absolute value decreases for x, we accept the shift. We then proceed with the next smallest $f_1 \circ \chi_1$. After we run through all outputs of f_1, we start over and continue with shifts 2α and so on.[10] Since we always shift in the direction of the mode, the distribution μ_1 will become more concentrated. The iteration stops if no further shifts are possible, since no further optimization is possible. For our tests, the mode remained the same under this procedure—to ensure this property, one can also only allow shifts that do not change the mode. Naturally, for a smaller step size, the final MAE is better, but the optimization takes longer. However, each LTE can decide how much time it can invest in optimizing the distribution. Figure 2 shows a construct μ for a binomially distributed $f_1 \circ \chi_i$. Furthermore, Fig. 5 shows that the mean absolute error decreases mostly independently of the size of the dataset. As in Section V, more extreme p in the underlying binomial distribution lead to a slower decrease.

E. Imprecise Knowledge of the input distribution

Recall that privacy for a prior distribution $\hat{\psi}_i$ implies privacy for $\psi_i := \hat{\psi}_i \circ \xi_i$ for any distribution ξ_i. This is not surprising given that the convolution corresponds to adding a sample from ξ_i, which cannot reduce privacy. In particular, the convolution with some ξ_i will make the distribution less concentrated, i.e., the distribution $\hat{\psi}_i$ provides more obfuscation than ψ_i. The LTE can use this effect by making a conservative guess $\hat{\psi}_i$ for an actual distribution ψ.[11]

Using a more concentrated distribution $\hat{\psi}_i$ naturally reduces the utility, since larger noise is employed by our mechanism. We have added an empirical evaluation in Fig. 6 that shows how the utility decreases if the LTE chooses instead of the real distribution ψ_i a similarly shaped distribution $\hat{\psi}_i$ with smaller variance, i.e., a more concentrated distribution. For our example of a binomial distribution $\psi_i = \hat{\psi}_i * \xi_i = \mathsf{Bin}(p_i, n_i)$ we chose $\hat{\psi}_i = \mathsf{Bin}(p_i, cn_i), \xi_i = \mathsf{Bin}(p_i, (1-c)n_i)$ such that the LTE uses $\hat{\psi}_i$ with $\frac{1}{n_i} \leq c < 1$ times smaller variance instead of the actual distribution ψ_i. Our results show that the utility decreases with smaller variance, but remains better than the utility of the standard geometric approach.

VII. Evaluation

To evaluate the privacy improvements of our solution, we compare the single-party case from Section V against the geometric mechanism [33] for the central model of DP and the multi-party case from Section VI against the local model of DP[12] in a realistic environment for our use case. As a reminder, our use case involves mail analytics in a corporate group with a holding company and several subsidiaries that are independent legal entities. The holding company uses the analytical data to visualize the corporate group's workflow and potentially propose reorganizations to optimize it. Our objective is to guarantee the privacy of the workers' emails.

For our benchmarks, we analyzed a real-world database from a large corporate group in telecommunications to determine an input distribution and sensitivity. For legal reasons, we did not run our benchmarks on the real employee data, but on a dataset [19] recreated from the input distributions and sensitivity, i.e., the used dataset comes with the same features as the real-world dataset. Since our mechanisms from Sections V and VI only use the input distribution and sensitivity, we get realistic benchmarks—for details, see the different instances of Alg. 1 described below.

The holding company has different options for querying the database, e.g., it can ask for statistics from all or only a part of the subsidiaries, or to only include e-mails from product managers to senior developers based in different locations in

[10] If $2\alpha > 1$ we assign the weight of A to $x + \lfloor 2\alpha \rfloor$ and $x + \lfloor 2\alpha \rfloor + 1$ and analogously for all other cases.

[11] Note that by choosing $\hat{\psi}_i$ sufficiently concentrated the LTE can account for possible malicious contributions, e.g. too large, from colluding employees or other factors that could affect the real input distribution.

[12] Recall that for the local model we use the SGDL distribution to sample noise since we split the geometric mechanism across the LTEs using its property of infinite divisibility as in [13], [18].

a given timeframe. In the following, we present benchmarks where the holding company randomly selected 10 subsidiaries. In our data set, each subsidiary employed between 50 and 400 workers. For the benchmarks, we selected a random subset of workers for each subsidiary, modeling queries $(f, f_1, \ldots, f_m)$ by the holding company that only address the correspondence between different job titles or locations. As before, we choose for f and f_i aggregation (in this case, over the random subset of subsidiaries and workers only). Let X_i be the data to these subsets of workers in LTE_i and χ_i the corresponding distribution (induced from the distribution on the whole dataset). In particular, χ_i provides the sensitivity Δ_{f_i} of the query, i.e., the maximal number of e-mails by an employee in the subset that has positive χ_i-probability. Let Δ_f be the maximal sensitivity occuring in any of the selected subsidiaries.

We apply Alg. 1 to the dataset corresponding to the query. For the single-party, multi-party case as well as the baselines from central and local DP, the functionalities Precomputation and Sampling in Alg. 1 are defined below.

Central model: The Precomputation functionality sets $D(X) = \mathsf{Geo}_{\varepsilon/\Delta_{f_1}}$. The Sampling functionality outputs $f_1(X_1) + e$, where $e \leftarrow D(X)$.

Our single party method: The Precomputation functionality runs the mechanism M from Section V with ε replaced by ε/Δ_{f_1}. For the benchmarks, we used Alg. 2 to compute $(r_{x,s})_{0\le x,s\le \Delta_{f_1}|X_1|}$.[13] Sampling samples as in Remark 8.

Local model: The Precomputation functionality sets $D(X_i) \sim \mathsf{SGDL}(\exp(-\varepsilon/\Delta_f), 1/10)$. The Sampling functionality outputs $f(f_i(X_i) + e_i)_{1\le i\le 10})$, where $e_i \leftarrow D(X_i)$.

Our multi-party method: The Precomputation functionality applies the mechanism from Section VI to X_i for each $1 \le i \le 10$. Following our discussion in Section VI we developed and ran [19] to compute a $(r_{x_i,s_i})_{x_i\in I_i, s_i\in I}$ for each LTE_i. The Sampling functionality uses Remark 8, i.e. it samples $s_i \leftarrow \mathsf{SGDL}(\exp(-\varepsilon/\Delta_f), 1/10)$ for $1 \le i \le 10$ and sets $\overline{s}_i = f_i(X_i) + s_i$ if $f(X_i) + s_i \notin I$, otherwise it samples $\overline{s}_i \leftarrow (c_i^{-1} r_{x_i,s_i})_{x_i\in I_i, s_i\in I}$ where $c_i = \sum_{s_i\in I} \mathsf{SGDL}(\exp(-\varepsilon/\Delta_f), 1/m)$ is a normalization. Finally, it outputs$f(f_1(X_1) + \overline{s}_1, \ldots, f_{10}(X_{10}) + \overline{s}_{10})$.

The results for our two mechanisms and the baselines are shown in Fig. 7. The Figure contains the average absolute and relative errors for all four mechanisms over 100 randomly selected queries and 10000 sampled datasets X_i.

The results show an improvement of both our methods compared to the central and local models. We get an improvement of 70% in the single-party case and of 15% in the multi-party case. Moreover, in Fig. 8, we can see how the utility changes with the distance to the mode. As expected, the closer the actual output $f(X_1)$ is to the mode, the smaller the absolute error with our mechanism. Nevertheless, even for extreme values, we never get worse than the geometric mechanism.

[13]Note that $x \in X_1$ and hence $\Delta_{f_1}|X_1|$ is an upper bound for the total number of e-mails in X_1, i.e. we can assume $0 \le x \le \Delta_{f_1}|X_1|$.

VIII. Conclusion and Future Work

In this work, we studied a multi-party setup where the parties, i.e., our LTEs, each possess a local dataset from different sources, i.e., our data owners, and together run a computation on these datasets that outputs a result to a possibly malicious central entity. We focused on computations consisting of a local precomputation by each party and interactive aggregation of the precomputation results, as is common in real-world use cases, e.g., in Federated Analytics or Learning.

The objective of our work was to protect the privacy of each single data owner, while improving the utility of the output compared to standard DP-approaches.

In Sections V and VI we proposed new DP-mechanisms that make use of a prior distribution of the inputs. We show that our new mechanisms improve over the state-of-the-art geometric mechanism theoretically. Moreover, we evaluate our mechanisms for a real-world use case in Section VII. Our results show that our new mechanisms come with improved utility compared to the geometric mechanism.

As we explained in Section I, our results apply beyond the specifics of the corporate use-case to all applications that fall within our setup outlined in Section IV, e.g., for Federated Learning (FL) applications as already discussed in Section I.

Applying our approach in FL is, however, more involved in general. While our mechanisms transfer to a single epoch of FL training directly, estimating the prior distribution used by our mechanisms is difficult with multiple epochs, since the input distributions of other LTEs affect the local training in consecutive epochs. Even if an LTE has access to the input (shadow) distribution of the other LTEs, what is often assumed in FL literature [17], [41], it is unlikely that there is an explicit relation to the resulting distribution of ML parameters after several epochs due to the complex stochastic nature of gradient descent. We therefore expect LTEs to rely on empirical methods, e.g., Monte Carlo simulation, to estimate the prior distribution. Using empirical methods is, however, typical in FL, e.g., when analyzing inference attacks [42]. Ultimately, we expect our mechanisms to come with similar utility gains for FL applications than in the corporate use case—we leave a detailed analysis for future work.

Acknowledgment

This research was supported by the German Federal Ministry of Education and Research (CRYPTECS under grant agreement 16KIS1441 and QCyber under grant agreement 16KIS2590K) and by the French National Research Agency (CRYPTECS under grant agreement ANR-20-CYAL-0006 and CHIST-ERA project PATTERN under grant agreement ANR-23-CHR4-0008), and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under grants 411720488 and 548713845. Andreas Athanasiou was also supported by the EU project RECITALS (EU grant agreement 101168490).

References

[1] A. R. Elkordy *et al.*, "Federated Analytics: A Survey," *APSIPA Trans. Signal Inf. Process*, vol. 12, no. 1, 2023.

[2] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *AISTATS 2017*, 2017.

[3] C. Dwork and A. Roth, "The Algorithmic Foundations of Differential Privacy," *FToCS 2014*, vol. 9, no. 3–4, 2014.

[4] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating Noise to Sensitivity in Private Data Analysis," in *TCC 2006*. Springer.

[5] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *ACM CCS '16*, 2016.

[6] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What Can We Learn Privately?" *SIAM JoC*, 2011.

[7] D. Chaum, C. Crépeau, and I. Damgård, "Multiparty Unconditionally Secure Protocols," in *ACM STOC '88*, 1988.

[8] B. Anandan and C. Clifton, "Laplace noise generation for two-party computational differential privacy," in *PST 2015*.

[9] S. Goryczka and L. Xiong, "A Comprehensive Comparison of Multiparty Secure Additions with Differential Privacy," *IEEE Transactions on Dependable and Secure Computing*, vol. 14, no. 5, 2017.

[10] C. Wei, R. Yu, Y. Fan, W. Chen, and T. Wang, "Securely Sampling Discrete Gaussian Noise for Multi-Party Differential Privacy," in *ACM CCS '23*, 2023.

[11] Y. Wei *et al.*, "Distributed Differential Privacy via Shuffling Versus Aggregation: A Curious Study," *Trans. Info. For. Sec.*, vol. 19.

[12] A. Cheu, A. Smith, J. Ullman, D. Zeber, and M. Zhilyaev, "Distributed Differential Privacy via Shuffling," in *EUROCRYPT 2019*. Springer.

[13] B. Ghazi, R. Kumar, P. Manurangsi, and R. Pagh, "Private Counting From Anonymous Messages: Near-Optimal Accuracy with Vanishing Communication Overhead," in *ICML 2020*.

[14] B. Balle, J. Bell, A. Gascón, and K. Nissim, "Private Summation in the Multi-Message Shuffle Model," in *ACM CCS '20*, 2020.

[15] R. Küsters, T. Truderung, and A. Vogt, "Verifiability, Privacy, Coercion-Resistance: New Insights from a Case Study," in *IEEE S&P 2011*.

[16] S. Biswas and C. Palamidessi, "Privic: A privacy-preserving method for incremental collection of location data," *PETS 2024*.

[17] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, " Membership Inference Attacks Against ML Models ," in *IEEE S&P 2017*.

[18] A. Athanasiou, K. Chatzikokolakis, and C. Palamidessi, "Enhancing Metric Privacy With a Shuffler," in *PETS 2025*.

[19] F. A. Escobar, A. Athanasiou, and P. Reisert, "Implementation to this paper," https://publ.sec.uni-stuttgart.de/csf-26-opt-dp-in-fa.zip, 2025.

[20] A. Bittau, "Prochlo: Strong Privacy for Analytics in the Crowd," in *ACM SOSP '17*, 2017.

[21] Ú. Erlingsson, V. Feldman, I. Mironov, A. Raghunathan, K. Talwar, and A. Thakurta, "Amplification by Shuffling: From Local to Central Differential Privacy via Anonymity," USA, 2019.

[22] D. Lu, T. Yurek, S. Kulshreshtha, R. Govind, A. Kate, and A. Miller, "HoneyBadgerMPC & AsynchroMix: Practical Asynchronous MPC and its Application to Anonymous Communication," in *ACM CCS '19*, 2019.

[23] I. Abraham, B. Pinkas, and A. Yanai, "Blinder–Scalable, Robust Anonymous Committed Broadcast," in *ACM CCS '20*.

[24] A. Kate, "RPM: Robust Anonymity at Scale," in *PETS 2023*.

[25] D. L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Commun. ACM*, vol. 24, no. 2, 1981.

[26] T. Haines and J. Müller, "Sok: Techniques for verifiable mix nets," in *IEEE CSF 2020*, 2020.

[27] I. Damgård, V. Pastro, N. Smart, and S. Zakarias, "Multiparty Computation from Somewhat Homomorphic Encryption," in *CRYPTO 2012*.

[28] D. Boneh, E. Boyle, H. Corrigan-Gibbs, N. Gilboa, and Y. Ishai, "Lightweight Techniques for Private Heavy Hitters," in *IEEE SP*, 2021.

[29] Y. Jia, S.-F. Sun, H.-S. Zhou, J. Du, and D. Gu, "Shuffle-based Private Set Union: Faster and More Secure," in *USENIX 2022*.

[30] R. Bhaskar, A. Bhowmick, V. Goyal, S. Laxman, and A. Thakurta, "Noiseless database privacy," in *ASIACRYPT 2011*.

[31] R. Hladík and J. Tětek, "Smooth sensitivity revisited: Towards optimality," 2024. [Online]. Available: https://arxiv.org/abs/2407.05067

[32] I. V. Linnik and I. V. Ostrovskiĭ, "Decomposition of random variables & vectors," in *Bul. of the American Math. Society*, vol. 48, no. 4, 1978.

[33] A. Ghosh, T. Roughgarden, and M. Sundararajan, "Universally utility-maximizing privacy mechanisms," in *STOC 2009*.

[34] S. Goryczka and L. Xiong, "A comprehensive comparison of multiparty secure additions with differential privacy," *IEEE TDSC*, 2017.

[35] S. V. Lekshmi and S. Sebastian, "A Skewed Generalized Discrete Laplace Distribution," *IJMSI*, vol. 2, no. 3, 2014.

[36] I. Damgård, M. Keller, E. Larraia, V. Pastro, P. Scholl, and N. P. Smart, "Practical Covertly Secure MPC for Dishonest Majority - Or: Breaking the SPDZ Limits," in *ESORICS 2013*.

[37] D. Mestel, J. Müller, and P. Reisert, "How Efficient are Replay Attacks against Vote Privacy?" in *IEEE CSF 2022*.

[38] M. B. Cohen, Y. T. Lee, and Z. Song, "Solving Linear Programs in the Current Matrix Multiplication Time," in *ACM SIGACT 2019*.

[39] M. A. Goberna, *Linear Semi-infinite Optimization: Recent Advances*. Springer US, 2005.

[40] J. J. Koenderink, "The Structure of Images," *Biological Cybernetics*, 1984.

[41] L. Bai, H. Hu, Q. Ye, H. Li, L. Wang, and J. Xu, "Membership inference attacks and defenses in federated learning: A survey," *ACM Comput. Surv.*, vol. 57, no. 4, 2024.

[42] M. Nasr, R. Shokri, and A. Houmansadr, "Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning," in *IEEE S&P 2019*.

[43] A. Pinkus, *Totally Positive Matrices*, ser. Cambridge Tracts in Mathematics. Cambridge University Press, 2009.

APPENDIX A
PROOFS AND ALGORITHMS

Lemma 3. *Let χ_1 be a distribution on $\mathbb{R}^n$ with support in I_1^n with $I_1 = \{a, \dots, b\}$. Let $w_j, w'_j \in I_1$ with $w_j < w'_j$ and $\chi^{w_j} \sim_c \chi^{w'_j}$. Then $\mathsf{Quot}[\mathsf{R}^j_{\mathsf{Geo},\chi}](s) = q^{w_j - w'_j}$ for $s \le a$ and $\mathsf{Quot}[\mathsf{R}^j_{\mathsf{Geo},\chi}](s) = q^{w'_j - w_j}$ for $s \ge b$. Furthermore, if $f_1 \circ \chi_1$ has no local minima (outside of the boundaries), then $\mathsf{Quot}[\mathsf{R}^j_{\mathsf{Geo},\chi}](s)$ is monotonically decreasing in s.*

Proof. W.l.o.g. we set $j = 1$. Note that $\mathsf{Pr}(\mathsf{R}^j_{\mathsf{Geo},\chi}(w_j) = s) = \sum_{s' \in f_1(w_j, I_1^{n-1})} \mathsf{Pr}(f_1(w_j, y) = s' | y \leftarrow \chi^{w_j}))\mathsf{Geo}_\varepsilon(s - s')$. By linearity of f_1, we can rewrite this term to $\sum_{s' \in f_1(w_j, I_1^{n-1})} \mathsf{Pr}(f_1(0, y) = s' - f_1(w_j, 0) | y \leftarrow \chi^{w_j})\mathsf{Geo}_\varepsilon(s - s') = \sum_{s' \in f_1(0, I_1^{n-1})} \mathsf{Pr}(f_1(0, \chi^{w_j}) = s')\mathsf{Geo}_\varepsilon(s - s' - f_1(w_j, 0))$.For $s \le a$, we get $s - s' - f_1(w_j, 0) \le 0$. But then $\mathsf{Geo}_\varepsilon(s - s' - f_1(w_j, 0))q = \mathsf{Geo}_\varepsilon(s - 1 - s' - f_1(w_j, 0))$ and the result on the quotient follows for $s \le a$. The case $s \ge b$ can be treated analogously.

For the second statement, we use Remark 6 to reduce to the case $w' = w + 1$. If we set $g_+ = \sum_{x \in \mathbb{N}: x < s} q^{s-x} f_1 \circ \chi_1^{w_j}(x)$ and $g_- = \sum_{x \in \mathbb{N}: x \ge s} q^{x-s} f_1 \circ \chi_1^{w_j}(x)$ then the DP-quotient becomes $\frac{g_+ + g_-}{g_+ q^{-1} + g_- q} = q^{-1} + \frac{g_+ - g_+ q^{-2}}{g_+ q^{-1} + g_- q} = q^{-1} - \frac{(q^{-1} - q)}{1 + q^2 g_- / g_+}$. Now, if s increases, then g_+ increases and g_- decreases in the limit $q \to 1$. Hence $1 + q^2 g_- / g_+$ decreases, and our quotient decreases as expected. □

Remark 9. Assume that the outputs of each LTE are distributed as $\mu_i * \mathsf{SGDL}_{1/m}$ as in Section VI, where LTE adds noise $\mathsf{SGDL}_{1/m}(y - s_1)$ to some modified output $A_{x,y}$ depending on the internal result x. Then (5) becomes an equality for $s \le a_1$. Now LTE_1 modifies its mechanism and outputs according to $r_{x,s_1} \neq \mathsf{SGDL}_{1/m}(x - s_1)$ for $x \in I_1$ such that $\sum_{x \in I_1} \omega_{q,1,w_j}^{f_1 \circ \chi_1}(x) r_{x,s_1} = h(s_1) + \sum_{x \in I_1} \omega_{q,1,w_j}^{f_1 \circ \chi_1}(x) \mathsf{SGDL}_{1/m}(x - s_1)$ increases by a non-trivial non-negative function $\Delta(s_1)$ under the condition that $\sum_{x \in I_1} \omega_{q,1,w_j}^{f_1 \circ \chi_1}(x) r_{x,s_1} \le 0$ still holds. This was our approach in Section V, where we used the non-tightness of the inequality to improve utility (cf. also Algorithm 2). Then the right-hand side of (5) increases by $\eta^\neg * \Delta$. Since $\eta^\neg$ is positive and Δ is non-negative (and non-trivial), this term has positive support

everywhere on $\mathbb{Z}$. In particular, the right-hand side of (5) is increased, and therefore (5) can no longer hold for $s \leq a_1$. Note that the non-negativity of Δ is not necessary to break privacy. Any function Δ that has large enough positive values for small s_1 increases the right-hand side of (5).

Remark 10. Recall from linear programming [38], [39] that optimal solutions are extreme points of the space defined by the constraints. In particular, they are boundary points and hence (some) constraints are satisfied tightly. It is therefore possible that an optimal solution satisfies (9) tightly. If this is the case, then (9) might already imply $\mu^\neg * u_q = 0$, which significantly simplifies the number of necessary constraints, e.g. (10) becomes redundant. For this implication to hold, it is, for example, sufficient that $\langle w_s := (\mathsf{SGDL}_\beta(s), \dots, \mathsf{SGDL}_\beta(s + a_1 - b_1)) : s \leq 0\rangle = \mathbb{R}^{b_1 - a_1 + 1}$, i.e., that vectors consisting of Laplace probabilities span the whole space. Since the resulting matrix $(w_0, \dots, w_{a_1 - b_1})$ is log-convex this matrix is for small β in fact totally-positive [43] and in particular invertible, which then shows $\mu^\neg * u_q = 0$. While we can currently not prove that an optimal solution satisfies (9) tightly or that as a result $\mu^\neg * u_q = 0$ for all $0 < \beta < 1$, our algorithm from Section VI-C already uses the tight constraint.

APPENDIX B
MAE IMPROVEMENT OF WEIGHT SHIFTING

In this appendix, we want to continue our description of our new noise mechanism from Section VI. Recall that we found two constraints for pairwise modifications by vector Δ, namely, $\sum_{x \in I_1} f \circ f_1 \circ \chi_1(x)\Delta(x) = 0 = \sum_{x \in I_1} \omega^{f_1 \circ \chi_1}_{q,1,w_j}(x)\Delta(x)$ or $\sum_{x \in I_1} f \circ f_1 \circ \chi_1(x)\Delta(x) = 0 = \sum_{x \in I_1} \omega^{f_1 \circ \chi_1}_{q^{-1},1,w_j}(x)\Delta(x)$ depending on whether $\eta^\neg(s - s_1) - \eta^\neg(s - s_2)$ is increasing or decreasing (if we choose $s_1 < s_2$).

Note that the conditions always ensure that a non-trivial solution vector Δ has positive and negative values. Since the rank of the linear problem is at most 2, we find solutions with three non-zero entries x_1, x_2, x_3 (for $|I_1| > 2$). By multiplying by -1, we can then assume that two entries are positive and one is negative. Finally, we can scale a solution such that $r_{x,s_1} + \Delta(x) \in [0, 1]$ and similarly for the other terms. Thus we can assume Δ satisfies $\Delta(x_1), \Delta(x_2) \geq 0 > \Delta(x_3)$.

We next investigate how our modification changes the absolute error. For $x \leq s_1 < s_2$ note that $r_{x,s_1} \leftarrow r_{x,s_1} - \Delta(x), r_{x,s_2} \leftarrow r_{x,s_2} + \Delta(x)$ increases the absolute error by $\Delta(x)|s_2 - x| - \Delta(x)|s_1 - x| = \Delta(x)(s_2 - s_1)$. We can keep the absolute error constant if we simultaneously shift $r_{x,s'_1} \leftarrow r_{x,s'_1} - \Delta'(x), r_{x,s'_2} \leftarrow r_{x,s'_2} + \Delta'(x)$ for some $x \leq s'_2 < s'_1$ and a suitable $\Delta'(x_1), \Delta'(x_2) \geq 0 > \Delta'(x_3)$, e.g. for $\Delta = \Delta'$ and $s'_1 + s_1 = s_2 + s'_2$. Assume from now on that s_1, s_2, s'_2, s'_1 and Δ, Δ' are such that the overall contribution to the MAE is 0 (see Alg. 5 for an explicit construction of suitable Δ, Δ'.)

We still need to determine the contribution of the x_3 component. Consider shifts of the form $x_1 \leq s_1 \leq s_2 \leq x_3 \leq s'_2 \leq s'_1 \leq x_2$. For these shifts the absolute error for x_1, x_2 stays the same (by construction), but for x_3, it changes by $\Delta(x_3)(x_3 - s_1) - \Delta(x_3)(x_3 - s_2) - \Delta'(x_3)(x_3 - s'_2) + \Delta'(x_3)(x_3 - s'_1) = \Delta(x_3)(s_2 - s_1) + \Delta'(x_3)(s'_1 - s'_2) < 0$. Note that privacy always holds for s'_1, s'_2 as long as we do not pass the mode and switch to the other set of constraints, i.e., s'_1, s'_2 both smaller or both larger than the mode of $\eta^\neg$.

We can now choose x_3 values for which the absolute error condition does not yet hold, and improve it. For x_1, x_2 we choose rows where the MAE condition is already satisfied, since the gain in absolute error by the construction above is for these rows 0. Once all conditions are satisfied, we can improve further to ultimately improving over the MAE of the geometric mechanism. Alg. 5 shows our approach for a binomial input distribution, where we can explicitly compute the solutions to our linear equations. Namely, for $f \circ f_1 \circ \chi_1 = \mathsf{Bin}(p, n), I_1 = \{0, \dots, n\}$ and conditional distribution $\mathsf{Bin}(p, n-1)$ the conditions become $\sum_{x \in I_1} \mathsf{Bin}(p, n-1)(x)\Delta(x) = 0 = \sum_{x \in I_1} \mathsf{Bin}(p, n)(x)\Delta(x)$. By choosing x_1, x_2, x_3 as before this becomes $\sum_{j=1}^3 w_j \Delta(x_j) = 0 = \sum_{j=1}^3 w_j^\neg \Delta(x_j)$ with $w_j = \mathsf{Bin}(p, n)(x_j), w_j^\neg = \mathsf{Bin}(p, n-1)(x_j)$. If $w_2 w_3^\neg - w_2^\neg w_3 \neq 0$, we find a solution $\Delta(x_1) \leftarrow 1, \Delta(x_2) \leftarrow \frac{w_3 w_1^\neg - w_3^\neg w_1}{w_2 w_3^\neg - w_2^\neg w_3}$ and $\Delta(x_3) \leftarrow -\frac{\Delta(x_1) w_1 + \Delta(x) w_2}{w_3}$. Re-scaling leads to valid probabilities and to a solution to all constraints (apart from the MAE constraint Eq. (8)). Other cases can be treated similarly. Note that in Alg. 5 we shift the weights stepwise with $s_2 = s_1 + 1, s'_2 + 1 = s'_1$, symmetrically ($s_1 = n - s'_1$) and use all possible choices $x_1 < s_1, s'_1 < s_2$ for optimization.

Let $V = \{x \in I_1 : \omega^{f_1 \circ \chi_1}_{q,1,w_j}(x) \leq 0 \leq \omega^{f_1 \circ \chi_1}_{q^{-1},1,w_j}(x)\}$;
Order the elements in V in a vector u such that $f_1 \circ \chi_1(u_j) \leq f_1 \circ \chi_1(u_{j+1})$ for $0 \leq j < |V|$;
`/* small index ⇒ low MAE impact. */`
$b_y \leftarrow 0, \forall y \in V$;
for $u_0 \in V$ **do**
  for $s \in [1, \dots, a_0[, s \neq u_0$ **do**
    $c_s = \sum_{y \in [0,\dots,a_0] \setminus V} \omega^{f_1 \circ \chi_1}_{q,1,w_j}(x) r_{x,s}$;
    $d_s = \sum_{x \in V} \omega^{f_1 \circ \chi_1}_{q,1,w_j}(x) r_{x,s}$;
    $c'_s = \sum_{y \in [0,\dots,a_0] \setminus V} \omega^{f_1 \circ \chi_1}_{q^{-1},1,w_j}(x) q r_{x,s}$;
    $d'_s = \sum_{x \in V} \omega^{f_1 \circ \chi_1}_{q,1,w_j}(x) q r_{x,s}$; `/* By definition: $d_s \leq 0$, $d'_s \geq 0$. */`
    Let $g_s \in [0, 1]$ be minimal with $c_s + d_s g_s \leq 0$, $c'_s + d'_s g_s \geq 0$; `/* ∃$g_s$, since` $c_s + d_s \leq 0$, $c'_s + d'_s \geq 0$ `for` Geo_ε. */
    Set $b_y \mathrel{+}= r_{y,s}(1 - g_s)$, $r_{y,s} \leftarrow r_{y,s} g_s$ for all $y \in V$; `/* Tight Eq. (1). */`
  Set $r_{u_0,u_0} \mathrel{+}= b_{u_0}$, $b_{u_0} \leftarrow 0$, $V \leftarrow V \setminus \{u_0\}$.; `/* Assigns remaining weight. */`
for $j = 1$ *to* $|V| - 1$ **do**
  Compute $g_{u_{j-1}}$ as before and set
  $b_y \mathrel{+}= (1 - g_{u_{j-1}}) r_{y,u_{j-1}}$,
  $r_{y,u_{j-1}} \leftarrow r_{y,u_{j-1}} g_{u_{j-1}}$ for all $y \in V$.
  $r_{u_j,u_j} \mathrel{+}= b_{u_j}$, $b_{u_j} \leftarrow 0$, $V \leftarrow V \setminus \{u_j\}$.

Algorithm 2: Compute distribution $(r_{x,s})_{x,s \in I_1}$ for our DP-mechanism for a single LTE.

Input: $x, \chi, I = \{a, \dots, b\}, A_{x,y}, \beta, \mathsf{shift}, \alpha$
Result: $\mu; (r_{x,s})_{x,s\in I}$
Let $\hat{\alpha} = \lfloor \alpha \mathsf{shift} \rfloor - \alpha \mathsf{shift}$;
for $y \in I$ **do**
  $A_{x,y} \leftarrow (1+\hat{\alpha})\delta_{x+\lfloor \alpha \mathsf{shift} \rfloor, y} - \hat{\alpha}\delta_{x+\lceil \alpha \mathsf{shift} \rceil, y}$;
Compute μ from $A_{x,y}$ (using input distribution χ),;
for $s = a$ **to** b **do**
  $r_{x,s} \leftarrow (1+\hat{\alpha})\mathsf{SGDL}_{1/\beta}(x + \lfloor \alpha \mathsf{shift} \rfloor - s)$, $r_{x,s} \mathrel{-}= \hat{\alpha}\mathsf{SGDL}_{1/\beta}(x + \lceil \alpha \mathsf{shift} \rceil - s)$;

Algorithm 3: Update $\mu, (r_{x,s})_{x,s\in I}$.

Input : $x, I = \{a, \dots, b\}, \alpha, \beta, (r_{x,s})_{x,s\in I}, A_{x,y}$
$E \coloneqq \max\{b - x, x - a\}$; $e' = \sum_{y\in I} A_{x,y}|x - y| \cdot r_{x,y}$;
for $1 \le e \le E$ **do**
  $e' \mathrel{+}= \sum_{y\in I} A_{x,y}(|x - y - e| \cdot r_{x,y+e} - |x - y + e| \cdot r_{x,y-e}) - 2 \cdot \mathsf{SGDL}_{\beta}(e)$;
output: $(e \le 0)$

Algorithm 4: Check Eq. (8).

Data: $\alpha, \beta, f_1 \circ \chi_1, 0 < \alpha \le 1$, $A_{x,y} \leftarrow \delta_{y,\lfloor (b-a)/2 \rfloor}, \mu(x) = \delta_{x,\lfloor (b-a)/2 \rfloor}$
Result: $(r_{x,s})_{x,s\in I}$
Let $\{a, \dots, b\}$ be the support of $f_1 \circ \chi_1$. Set $\psi(x) = f_1 \circ \chi_1(x + a)$ and $I = \{0, \dots, c\}$ for $c \leftarrow b - a$. Update $\mu, (r_{x,s})_{x,s\in I}$ with Alg. 3 for input $x, \psi, I, A_{x,y}, \beta, \alpha, \mathsf{shift} \leftarrow \alpha^{-1}(\lfloor c/2 \rfloor - x)$; /* Initialize */
for $\mathit{shift} = 0$ **to** $\alpha^{-1}\lfloor c/2 \rfloor$ **do**
  for $0 \le x \le \lfloor c/2 \rfloor$ **do**
    if *the MAE condition is satisfied for x (using Alg. 4)* **then break**;
    else
      Set $\mathsf{shift}_1 \leftarrow \alpha^{-1}(\lfloor c/2 \rfloor - x) - \mathsf{shift}$. Update $\mu, (r_{x,s})_{x,s\in I}$ with Alg. 3 for input $x, f_1 \circ \chi_1^0, I, A_{x,y}, \beta, \mathsf{shift}_1, \alpha$; /* Reduce shift until all conditions are met. */
    for $0 \le s < x$ **do**
      $\tilde{s}_1 \leftarrow x - (s+1), \tilde{s}_2 \leftarrow x - s$; $\hat{s}_1 \leftarrow c - x + s + 1, \hat{s}_2 \leftarrow c - x + s$; /* Shift to the diagonal. */
      for $0 \le y < \tilde{s_1}$ **do**
        for $(x_1, x_2, x_3, s_1, s_2, r, \Delta) \in \{(y, c-y, x, \tilde{s}_1, \tilde{s}_2, \tilde{r}, \tilde{\Delta}), (c-y, y, c-x, \hat{s}_1, \hat{s}_2, \hat{r}, \hat{\Delta})\}$ **do**
          $w_k \leftarrow \psi(x_k), w_k^0 \leftarrow \psi^0(x_k), \Delta(x_k) \leftarrow 0$ for $k = 1, 2, 3$;
          if $w_2 w_3^0 - w_2^0 w_3 \ne 0$ **then**
            $\Delta(x_1) \leftarrow 1$, $\Delta(x_2) \leftarrow \frac{w_3 w_1^0 - w_3^0 w_1}{w_2 w_3^0 - w_2^0 w_3}$;
          $\Delta(x_3) \leftarrow -\frac{\Delta(x_1)\cdot w_1 + \Delta(x_2)\cdot w_2}{w_3}$; /* Ensures output distribution/privacy. */
          $r \leftarrow \max\{t \ge 0 \colon \Delta(x_k)t + r_{x_k,s_1}, r_{x_k,s_2} - \Delta(x_k)t \in [0,1], \forall k = 1,2,3\}$; /* Probabilities */
        if $\tilde{r}\tilde{\Delta}(x_1) < \hat{r}\hat{\Delta}(x_1)$ **then**
          $\hat{r} \leftarrow \frac{\tilde{r}\tilde{\Delta}(x_1)}{\hat{\Delta}(x_1)}$; /* Ensures that MAE for x_1 does not increase. */
        if $\tilde{r}\tilde{\Delta}(x_2) > \hat{r}\hat{\Delta}(x_2)$ **then**
          $\tilde{r} \leftarrow \frac{\hat{r}\hat{\Delta}(x_2)}{\tilde{\Delta}(x_2)}$; /* Ensures that MAE for x_2 does not increase. */
        for $(x_1, x_2, x_3, s_1, s_2, r, \Delta) \in \{(y, c-y, x, s_1, s_2, \tilde{r}, \tilde{\Delta}), (c-y, y, c-x, \hat{s}_1, \hat{s}_2, \hat{r}, \hat{\Delta})\}$ **do**
          $r_{x_k,s_1} \mathrel{+}= \Delta(x_k)r, r_{x_k,s_2} \mathrel{-}= \Delta(x_k)r, \quad \forall k = 1,2,3;$ /* Update probabilities */

Algorithm 5: Computes the noise distribution with step size α if the two sets of constraints, i.e. s_1, s_2 smaller or larger the mode of $\eta^{\neg}$, are equivalent. This is the case for the considered type of input distribution, e.g. a binomial distribution.