Revealing the True Cost of Locally Differentially Private Protocols: An Auditing Perspective

Héber H. Arcolezi, Sébastien Gambs

▶ **To cite this version:**

Héber H. Arcolezi, Sébastien Gambs. Revealing the True Cost of Locally Differentially Private Protocols: An Auditing Perspective. Proceedings on Privacy Enhancing Technologies, 2024, 2024 (4), pp.123 - 141. 10.56553/popets-2024-0110 . hal-04644975

HAL Id: hal-04644975
https://inria.hal.science/hal-04644975

Submitted on 11 Jul 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Revealing the True Cost of Locally Differentially Private Protocols: An Auditing Perspective

Héber H. Arcolezi
Inria Centre at the University Grenoble Alpes
France
heber.hwang-arcolezi@inria.fr

Sébastien Gambs
Université du Québec à Montréal (UQAM)
Canada
gambs.sebastien@uqam.ca

ABSTRACT

While the existing literature on Differential Privacy (DP) auditing predominantly focuses on the centralized model (*e.g.*, in auditing the DP-SGD algorithm), we advocate for extending this approach to audit Local DP (LDP). To achieve this, we introduce the LDP-Auditor framework for empirically estimating the privacy loss of locally differentially private mechanisms. This approach leverages recent advances in designing privacy attacks against LDP frequency estimation protocols. More precisely, through the analysis of numerous state-of-the-art LDP protocols, we extensively explore the factors influencing the privacy audit, such as the impact of different encoding and perturbation functions. Additionally, we investigate the influence of the domain size and the theoretical privacy loss parameters ϵ and δ on local privacy estimation. In-depth case studies are also conducted to explore specific aspects of LDP auditing, including distinguishability attacks on LDP protocols for longitudinal studies and multidimensional data. Finally, we present a notable achievement of our LDP-Auditor framework, which is the discovery of a bug in a state-of-the-art LDP Python package. Overall, our LDP-Auditor framework as well as our study offer valuable insights into the sources of randomness and information loss in LDP protocols. These contributions collectively provide a realistic understanding of the local privacy loss, which can help practitioners in selecting the LDP mechanism and privacy parameters that best align with their specific requirements. We open-sourced LDP-Auditor in [4].

KEYWORDS

Local differential privacy, Privacy auditing, Privacy attacks.

1 INTRODUCTION

Differential Privacy (DP) [29] is now widely recognized as the gold standard for providing formal guarantees on the privacy level achieved by an algorithm. One of its extension, known as Local DP (LDP) [28, 39], aims at tackling the trust challenges associated with relying on a centralized server, such as those highlighted by various data breaches [50] and instances of data misuse [71]. In LDP, each user perturbs their own data locally before sharing it with a data aggregator or a central server. The fundamental idea behind LDP is to introduce carefully calibrated noise to the data to ensure individual privacy guarantees while allowing meaningful statistical analysis to be performed on the aggregated noisy data.

This work is licensed under the Creative Commons Attribution 4.0 International License. To view a copy of this license visit https://creativecommons.org/licenses/by/4.0/ or send a letter to Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.
Proceedings on Privacy Enhancing Technologies 2024(4), 123–141
© 2024 Copyright held by the owner/author(s).
https://doi.org/10.56553/popets-2024-0110

Formally, a randomized algorithm $\mathcal{M}$ satisfies (ϵ, δ)-local differential privacy ((ϵ, δ)-LDP), for $\epsilon \geq 0$ and $0 \leq \delta \leq 1$, if for any pair of input values $v_1, v_2 \in \text{Domain}(\mathcal{M})$ and all possible sets of outputs $O \subseteq \text{Range}(\mathcal{M})$, the following inequality holds:

$$\Pr[\mathcal{M}(v_1) \in O] \leq e^{\epsilon} \cdot \Pr[\mathcal{M}(v_2) \in O] + \delta. \tag{1}$$

In particular, (ϵ, δ)-LDP is also called approximate LDP, with the special case of $\delta = 0$ being called pure ϵ-LDP. On the one hand, an (L)DP mechanism is accompanied by the mathematical proof in Equation (1) that establishes a **theoretical upper bound** for the privacy loss, represented by the privacy parameters ϵ and δ. In particular, lower values of ϵ indicate stronger privacy guarantees. On the other hand, the recent and emerging field of DP auditing (*e.g.*, see [3, 19, 36, 40, 44, 48, 53, 54, 56, 58, 60]) aims at estimating an **empirical lower bound** for the privacy loss, denoted as ϵ_{emp}.

The role of DP auditing is crucial because it bridges the gap between theoretical guarantees and practical implementations, especially when theoretical bounds on privacy loss might be overly pessimistic or not sufficiently tight (*e.g.*, as in Differentially Private Stochastic Gradient Descent – DP-SGD [1]). In other words, DP auditing helps in understanding how well privacy-preserving mechanisms perform under different conditions and attack scenarios [54]. Furthermore, auditing can uncover potential vulnerabilities or flaws in the implementation that might not be apparent through theoretical analysis alone [27, 60, 61]. From a practical standpoint, the empirical estimation of the privacy loss through realistic attackers can also help practitioners make informed decisions and understand the implications of specific privacy parameter choices. These instances underscore the significance of empirically estimating and verifying the claimed privacy levels of (L)DP mechanisms.

1.1 Our Contributions

With these motivations in mind, in this paper, we introduce the LDP-Auditor framework, which is designed to audit LDP frequency estimation protocols and estimate their empirical privacy loss. Frequency (or histogram) estimation is a primary objective of LDP as it is a building block for more complex tasks. *This means our audit results are applicable and relevant to numerous tasks under LDP guarantees*, such as heavy hitter estimation [15, 68], joint distribution estimation [24, 41, 57, 75], frequent item-set mining [67, 72], machine learning [49, 74], frequency estimation of multidimensional data [5, 55, 64] and frequency monitoring [7, 10, 26, 32, 63].

More precisely, LDP-Auditor relies on Monte Carlo methods to estimate the probabilities $\hat{p}_0 = \Pr[\mathcal{M}(v_1) \in O]$ and $\hat{p}_1 = \Pr[\mathcal{M}(v_2) \in O]$ from Equation (1) through attacks. From this, an empirical privacy loss is computed, $\epsilon_{emp} = \ln\left((\hat{p}_0 - \delta)/\hat{p}_1\right)$, thus providing an estimate of the algorithm's privacy leakage. A comprehensive

discussion on the LDP-Auditor framework, including its detailed methodology and applications, is deferred to Section 4.

Unlike traditional DP-SGD auditing, in which the focus is on distinguishing neighboring datasets, LDP-Auditor assesses the distinguishability of inputs directly. To achieve this, we instantiate LDP-Auditor with distinguishability attacks based on recent adversarial analysis of LDP frequency estimation protocols [9, 31]. These attacks allow an adversary's to predict the user's input value based on the obfuscated output, enabling LDP-Auditor to directly evaluate the privacy guarantees offered by LDP mechanisms, making it well-suited for privacy auditing. In this context, expanding beyond [9, 31], we also introduce novel distinguishability attacks tailored to four additional LDP frequency estimation protocols based on histogram encoding [66], as well as general distinguishability attacks on LDP protocols for longitudinal studies (see Algorithm 2) and on LDP protocols for multidimensional data (see Algorithm 3).

As an example, Figure 1 illustrates an instance of our auditing results for a theoretical upper bound of $\epsilon = 2$ (indicated by the dashed red line) across eight ϵ-LDP frequency estimation protocols: Generalized Randomized Response (GRR) [37], Subset Selection (SS) [65, 73], Symmetric Unary Encoding (SUE) [32], Optimal Unary Encoding (OUE) [66], Thresholding with Histogram Encoding (THE) [66], Summation with Histogram Encoding (SHE) [29], Binary Local Hashing (BLH) [15] and Optimal Local Hashing (OLH) [66]. Among all these protocols, GRR demonstrated a tight empirical privacy loss estimation for ϵ_{emp} as it does not require a specific encoding. On the other hand, other LDP protocols presented ϵ_{emp} within $\leq$ 2x of the theoretical ϵ (such as SUE, THE and SHE), and even within $\leq$ 4x of the theoretical ϵ (like BLH). *These results indicate that either the state-of-the-art attacks are still not representative of the worst-case scenario or that the upper bound analyses of these LDP protocols are not tight. The latter assumption might occur for LDP protocols that incorporate sources of randomness (*e.g.*, due to hashing [2, 15, 59, 66])* not captured in the worst-case definition of LDP in Equation (1).

More specifically, ***we have investigated several factors influencing the audit***, including the effect of theoretical privacy loss parameters (ϵ and δ) in low, mid and high privacy regimes as well as the impact of the domain size k on local privacy estimation. ***Our investigation included detailed case studies to further explore specific facets of LDP auditing.*** Notably, our analysis assessed how variations in δ affect the empirical privacy loss, ϵ_{emp}, for approximate LDP variants [69] of the GRR, SUE, BLH and OLH protocols, alongside with the Gaussian Mechanism (GM) [30] and the Analytic GM (AGM) [14]. Moreover, given that BLH exhibited the least tight empirical privacy loss estimation ϵ_{emp}, we investigated the privacy loss of local hashing without LDP obfuscation. In addition, we examined the degradation of the empirical local privacy loss in repeated data collections compared to the theoretical upper bound imposed by the (L)DP sequential composition [30]. In this context, within a generic framework, we proposed distinguishability attacks on LDP protocols in *longitudinal studies* (*cf.* Algorithm 2). Furthermore, we addressed the case of *multidimensional data*, proposing distinguishability attacks for LDP protocols following the RS+FD [5] solution (*cf.* Algorithm 3). We also show

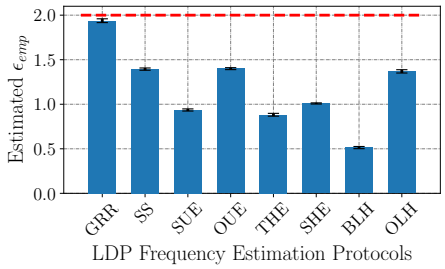


Figure 1: Comparison of estimated privacy loss ϵ_{emp} with theoretical upper bound $\epsilon = 2$ for eight pure LDP frequency estimation protocols. The dashed red line corresponds to the certifiable upper bound. While GRR closely aligns with the theoretical bound, others exhibit empirical ϵ_{emp} within $\leq$ 2x (*e.g.*, SUE) or even $\leq$ 4x (*i.e.*, BLH) of the theoretical ϵ value.

how *LDP-Auditor successfully identified a bug in one state-of-the-art LDP Python package*, in which the empirical privacy loss ϵ_{emp} contradicts the theoretical upper bound ϵ (see Figure 8).

Taking all these aspects into account, the coverage of our analysis is broadened, allowing for a more comprehensive assessment of the robustness of various LDP protocols in realistic data collection scenarios. More specifically, our main contributions in this paper can be summarized as follows:

- We introduce the LDP-Auditor framework, which aims to estimate the empirical privacy loss of LDP frequency estimation protocols. This framework provides a realistic assessment of privacy guarantees, which is essential for making informed decisions about LDP parameter selection and on stimulating the research of new privacy attacks.
- We introduce novel distinguishability attacks specifically tailored to LDP protocols for longitudinal studies and multidimensional data. These new attacks enrich the privacy analysis techniques available for examining the robustness of LDP mechanisms in practical settings.
- We conduct an extensive audit of various LDP protocols, analyzing the impact of factors such as privacy regimes, domain size and multiple data collections. This comprehensive analysis provides valuable insights into the resilience and effectiveness of nine state-of-the-art LDP mechanisms, fundamental building blocks for applications such as frequency monitoring [10, 26, 32, 63], heavy hitter estimation [15, 68] and machine learning [49, 74].
- We demonstrate the bug detection capabilities of LDP-Auditor by identifying an issue in a state-of-the-art LDP Python package. This highlights the practical significance of our framework in validating LDP implementations.

2 RELATED WORK

Differential privacy auditing, as introduced by Jagielski et al. [36], involves employing various techniques to empirically assess the extent of privacy leakage in machine learning algorithms through estimating the ϵ_{emp} privacy loss. These techniques are particularly

valuable when known analytical bounds on the DP loss lack precision, allowing for empirical measurements of privacy in such cases. For instance, DP auditing has been extensively investigated in evaluating the mathematical analysis for the well-known DP-SGD algorithm proposed by Abadi et al. [1]. The research literature on DP-SGD auditing covers both centralized [19, 36, 44, 53, 54, 56, 58, 60] and federated [3, 48] learning settings. Beyond privacy-preserving machine learning, privacy auditing has also been studied for standard DP algorithms [11, 17, 27, 34, 45]. For instance, some of these works consider a fully black-box scenario (*i.e.*, unknown DP mechanism) with the goal of estimating the ϵ-(L)DP guarantee provided [11, 34, 45]. Another line of research [17, 27] has been tailored to identify errors in algorithm analysis or code implementations, especially when derived lower bounds contradict theoretical upper bounds. While the works in [17, 27] could also be used to certify the ϵ-LDP guarantee through Monte Carlo estimations, our work considers realistic privacy attacks to LDP mechanisms to empirically estimate the privacy loss ϵ_{emp}. In other words, they would be able to answer "*is the claimed ϵ-LDP correct in this code implementation?*", whereas we alternatively answer "*is the claimed ϵ-LDP worst-case guarantee tight under state-of-the-art attacks?*".

This distinction highlights our emphasis on assessing the tightness of privacy guarantees under stringent adversarial conditions. Consequently, we envision our auditing analysis as an stimulus for advancing the current state-of-the-art in privacy attacks on LDP protocols and achieve tight empirical estimates for ϵ_{emp}. In this context, the existing literature on privacy attacks on LDP comprises several categories: (1) Distinguishability attacks [9, 21, 31] (adopted in this work), which enable adversaries to predict the users' input based on the obfuscated outputs; (2) Pool inference attacks [33], allowing adversaries to deduce a user's preferences or attributes from the aggregated data, such as inferring a user's preferred skin tone used in emojis; (3) Re-identification attacks [9, 52], aiming to uniquely identify a specific user within a larger population; and (iv) Attacks on iterative data collections [10, 35], which allows adversaries to detect a pattern change in longitudinal studies, such as when someone starts a diet by monitoring calorie consumption.

3 LDP FREQUENCY ESTIMATION PROTOCOLS

In this section, we review the necessary notation (*cf.* Table 1 in Appendix A) and background information of the LDP frequency estimation protocols. Throughout the paper, let $[n] = \{1, 2, \ldots, n\}$ denote a set of integers and $V = \{v_1, \ldots, v_k\}$ represent a sensitive attribute with a discrete domain of size $k = |V|$. We consider a distributed setting with n users and one untrusted server collecting the data reported by these users. The fundamental premise of (ϵ, δ)-LDP, as stated in Equation (1), is that the input to $\mathcal{M}$ cannot be confidently determined from its output, with the level of confidence determined by e^ϵ and δ. Therefore, the user's privacy is considered compromised if the adversary can correctly predict the user's value.

In recent works [9, 31], the authors introduced **distinguishability attacks** $\mathcal{A}$ to state-of-the-art LDP frequency estimation protocols. These attacks enable an adversary to predict the users' value $\hat{v} = \mathcal{A}(y)$, in which $y = \mathcal{M}(v)$ represents the reported value obtained through the ϵ-LDP protocol. In essence, although each LDP protocol employs different encoding and perturbation functions, the adversary's objective remains the same, namely to predict the user's true value by identifying the most likely value that would have resulted in the reported value y. The notion of distinguishability attacks provides a unified approach to evaluate the privacy guarantees offered by different LDP protocols.

We now provide a brief overview of state-of-the-art pure and approximate LDP frequency estimation protocols $\mathcal{M}$, along with their respective distinguishability attacks denoted as $\mathcal{A}_\mathcal{M}$. The attack $\mathcal{A}_\mathcal{M}$ generally relies on a "support set" [66], denoted as $\mathbb{1}_\mathcal{M}$, which is built upon the reported value y. The combination of these protocols and attack strategies will enable us to comprehensively audit the empirical privacy level provided by various LDP mechanisms.

3.1 Pure ϵ-LDP Protocols

Generalized Randomized Response (GRR). The GRR [37] mechanism generalizes the randomized response surveying technique proposed by Warner [70] for $k \geq 2$ while satisfying ϵ-LDP. Given a value $v \in V$, GRR(v) outputs the true value v with probability p, and any other value $v' \in V \setminus \{v\}$, otherwise. More formally:

$$\Pr[\text{GRR}(v) = y] = \begin{cases} p = \frac{e^\epsilon}{e^\epsilon + k - 1} \text{ if } y = v, \\ q = \frac{1}{e^\epsilon + k - 1} \text{ if } y \neq v, \end{cases} \tag{2}$$

in which $y \in V$ is the perturbed value sent to the server. The support set for GRR is simply $\mathbb{1}_{\text{GRR}} = \{y\}$. From Equation (2), $\Pr[y = v] > \Pr[y = v']$ for all $v' \in V \setminus \{v\}$. Therefore, the attack strategy $\mathcal{A}_{\text{GRR}}$ is to predict $\hat{v} = y$ [9, 31].

Subset Selection (SS). The SS [65, 73] mechanism was proposed for the case in which the obfuscation output is a subset of values Ω of the original domain V. The optimal subset size that minimizes the variance is $\omega = |\Omega| = \max\left(1, \left\lfloor \frac{k}{e^\epsilon + 1} \right\rceil\right)$. Given an empty subset Ω, the true value v is added to Ω with probability $p = \frac{\omega e^\epsilon}{\omega e^\epsilon + k - \omega}$. Finally, values are added to Ω as follows:

- If $v \in \Omega$, then $\omega - 1$ values are sampled from $V \setminus \{v\}$ uniformly at random (without replacement) and are added to Ω;
- If $v \notin \Omega$, then ω values are sampled from $V \setminus \{v\}$ uniformly at random (without replacement) and are added to Ω.

Afterward, the user sends the subset Ω to the server. The support set for SS is the subset of all values in Ω, *i.e.*, $\mathbb{1}_{\text{SS}} = \{v | v \in \Omega\}$. Therefore, the attack strategy $\mathcal{A}_{\text{SS}}$ is to predict $\hat{v} = \text{Uniform}(\mathbb{1}_{\text{SS}})$ [9, 31].

Unary Encoding (UE). UE protocols [32, 66] encode the user's input data $v \in V$, as a one-hot k-dimensional vector before obfuscating each bit independently. More precisely, let $\mathbf{v} = [0, \ldots, 0, 1, 0, \ldots, 0]$ be a binary vector with only the bit at the position v set to 1 while the other bits are set to 0. The obfuscation function of UE mechanisms randomizes the bits from $\mathbf{v}$ independently to generate $\mathbf{y}$ as follows:

$$\forall i \in [k]: \quad \Pr[\mathbf{y}_i = 1] = \begin{cases} p, \text{ if } \mathbf{v}_i = 1, \\ q, \text{ if } \mathbf{v}_i = 0, \end{cases} \tag{3}$$

in which $\mathbf{y}$ is sent to the server. There are two variations of UE mechanisms: (i) Symmetric UE (SUE) [32] that selects $p = \frac{e^{\epsilon/2}}{e^{\epsilon/2} + 1}$ and $q = \frac{1}{e^{\epsilon/2} + 1}$ in Equation (3), such that $p + q = 1$; and (ii) Optimal UE (OUE) [66] that selects $p = \frac{1}{2}$ and $q = \frac{1}{e^\epsilon + 1}$ in Equation (3). With

$\mathbf{y}$, the adversary can construct the subset of all values $v \in V$ that are set to 1, *i.e.*, $\mathbb{1}_{\text{UE}} = \{v|\mathbf{y}_v = 1\}$. There are two possible attack strategies $\mathcal{A}_{\text{UE}}$ [9, 31]:

- $\mathcal{A}^0_{\text{UE}}$ is a random choice $\hat{v} = \text{Uniform}([k])$, if $\mathbb{1}_{\text{UE}} = \emptyset$;
- $\mathcal{A}^1_{\text{UE}}$ is a random choice $\hat{v} = \text{Uniform}(\mathbb{1}_{\text{UE}})$, otherwise.

Local Hashing (LH). LH protocols [15, 66] use hash functions to map the input data $v \in V$ to a new domain of size $g \geq 2$, and then apply GRR to the hashed value. Let $\mathscr{H}$ be a universal hash function family such that each hash function $\text{H} \in \mathscr{H}$ hashes a value $v \in V$ into $[g]$ (*i.e.*, $\text{H} : V \to [g]$). There are two variations of LH mechanisms: (i) Binary LH (BLH) [15] that just sets $g = 2$, and (ii) Optimal LH (OLH) [66] that selects $g = \lfloor e^{\epsilon} + 1 \rceil$. Each user first selects a hash function $\text{H} \in \mathscr{H}$ at random and obfuscates the hash value $h = \text{H}(v)$ with GRR. In particular, the LH reporting mechanism is $\text{LH}(v) \coloneqq \langle \text{H}, \text{GRR}(h) \rangle$, in which $\text{GRR}(h)$ is given in Equation (2) while operating on the new domain $[g]$. Each user reports the hash function and obfuscated value $\langle \text{H}, y \rangle$ to the server. With these elements, the adversary can construct the subset of all values $v \in V$ that hash to y, *i.e.*, $\mathbb{1}_{\text{LH}} = \{v|\text{H}(v) = y\}$. There are two possible attack strategies $\mathcal{A}_{\text{LH}}$ [9, 31]:

- $\mathcal{A}^0_{\text{LH}}$ is a random choice $\hat{v} = \text{Uniform}([k])$, if $\mathbb{1}_{\text{LH}} = \emptyset$;
- $\mathcal{A}^1_{\text{LH}}$ is a random choice $\hat{v} = \text{Uniform}(\mathbb{1}_{\text{LH}})$, otherwise.

Histogram Encoding (HE). HE protocols [66] encode the user value as a one-hot k-dimensional histogram, $\mathbf{v} = [0.0, 0.0, \ldots, 1.0, 0.0, \ldots, 0.0]$ in which only the v-th component is 1.0. To satisfy ϵ-LDP, $\text{HE}(\mathbf{v})$ perturbs each bit of $\mathbf{v}$ independently using the Laplace mechanism [29]. Two different input values $v_1, v_2 \in V$ will result in two vectors with L1 distance of $\Delta_1 = 2$. Thus, HE will output $\mathbf{y}$ such that $\mathbf{y}_i = \mathbf{v}_i + \text{Lap}\left(\frac{\Delta_1}{\epsilon}\right)$. *In this paper, we propose distinguishability attacks on two pure ϵ-LDP HE protocols:*

- **Summation with HE (SHE)** [29]. With SHE, there is no post-processing of $\mathbf{y}$. Instead of constructing a support set, we describe our attacking strategy to SHE as follows. Let $P_V(v)$ be the prior probability of input value v, and let $P_Y(\mathbf{y}|v)$ be the likelihood of observing $\mathbf{y}$ given the true input value v. By the Bayes' theorem, the posterior probability of input value v given the observed $\mathbf{y}$ is:

$$P_V(v|\mathbf{y}) = \frac{P_Y(\mathbf{y}|v)P_V(v)}{\sum_{i=1}^{k} P_Y(\mathbf{y}|i)P_V(i)}. \tag{4}$$

  We can compute the likelihood $P_Y(\mathbf{y}|v)$ as follows. For a given v, the corresponding one-hot encoded histogram is $\mathbf{v}$. The reported value $\mathbf{y}$ is the sum of $\mathbf{v}$ and noise from a Laplace distribution with scale $b = 2/\epsilon$. Therefore, the likelihood of observing $\mathbf{y}$ given $\mathbf{v}$ is:

$$P_Y(\mathbf{y}|\mathbf{v}) = \frac{1}{(2b)^k} \exp\left(-\frac{|\mathbf{y} - \mathbf{v}|_1}{b}\right), \tag{5}$$

  in which $|\mathbf{y} - \mathbf{v}|_1$ is the L1 distance between $\mathbf{y}$ and $\mathbf{v}$. To perform the attack, we compute the posterior probability $P_V(v|\mathbf{y})$ for each possible input value $v \in V$ and output the most probable input value. In other words, given the reported $\mathbf{y}$, our Bayes optimal attack $\mathcal{A}_{SHE}$ outputs:

$$\hat{v} = \arg\max_{v \in V} P_V(v|\mathbf{y}). \tag{6}$$

  Note that this attack requires knowledge of the prior probability distribution $P_V(v)$. If the prior is unknown (assumed in this paper), one can use a uniform prior.
- **Thresholding with HE (THE)** [66]. With THE, the server (or the user) can construct the support set as $\mathbb{1}_{\text{THE}} = \{v \mid \mathbf{y}_v > \theta\}$, *i.e.*, each noise count whose value $> \theta$. The optimal threshold value for θ that minimizes the protocol's variance is within $(0.5, 1)$. With $\mathbb{1}_{\text{THE}} = \{v \mid \mathbf{y}_v > \theta\}$, we propose an adversary $\mathcal{A}_{\text{THE}}$ with two attack strategies:
  - $\mathcal{A}^0_{\text{THE}}$ is a random choice $\hat{v} = \text{Uniform}([k])$, if $\mathbb{1}_{\text{THE}} = \emptyset$;
  - $\mathcal{A}^1_{\text{THE}}$ is a random choice $\hat{v} = \text{Uniform}(\mathbb{1}_{\text{THE}})$, otherwise.

3.2 Approximate (ϵ, δ)-LDP Protocols

In this section, we describe two (ϵ, δ)-LDP protocols, which are based on the Gaussian mechanism [14, 30]. We defer the descriptions of approximate (ϵ, δ)-LDP variants [69] of GRR, SUE and LH protocols – namely, Approximate GRR (AGRR), Approximate SUE (ASUE), Approximate LH (ALH) – to Appendix B.

HE with Gaussian Mechanism (HE-GM) [14, 30]. Similar to HE protocols of Section 3.1, HE-GM protocols encode the user value as a one-hot k-dimensional histogram. Then, HE-GM($\mathbf{v}$) perturbs each bit of $\mathbf{v}$ independently using a Gaussian mechanism (GM) [14, 30]. Two different input values $v_1, v_2 \in V$ will result in two vectors with L2 distance of $\Delta_2 = \sqrt{2}$. Thus, HE-GM will output $\mathbf{y}$ such that $\mathbf{y}_i = \mathbf{v}_i + \mathcal{N}\left(0, \sigma^2\right)$, in which σ is determined by $\epsilon, \delta,$ and Δ_2. When using the well-established GM for $\epsilon, \delta \in (0, 1)$, $\sigma = \frac{\Delta_2}{\epsilon}\sqrt{2\ln(1.25/\delta)}$ [30]. In this paper, we also consider the Analytic GM (AGM) [14], which is an improved version of the GM [30] and can be applied for any $\epsilon > 0$. The main difference between GM and AGM is the method to parameterize σ. With AGM, σ is calculated analytically as demonstrated in [14, Algorithm 1] and its implementation [12]. Hereafter, we will specifically denote "AGM" and "GM" when referring to HE-GM instantiated with AGM and GM, respectively.

Building upon our distinguishability attack's description of the SHE protocol with Laplace noise, we extend the attack analysis to HE-GM protocols. The overall strategy, including the use of Bayes' theorem to compute posterior probabilities, remains consistent with our prior description in Section 3.1 (*cf.* Equation (4)). However, the key difference lies in the noise distribution used for ensuring LDP.

While the Laplace mechanism involves adding noise drawn from a Laplace distribution with scale $b = 2/\epsilon$, the Gaussian mechanism adds noise following the normal distribution, (*i.e.*, $\mathcal{N}\left(0, \sigma^2\right)$). This needs a different computation for the likelihood $P_Y(\mathbf{y}|v)$ in Equation (5), reflecting the properties of Gaussian noise. Accordingly, the likelihood of observing $\mathbf{y}$ given $\mathbf{v}$ under Gaussian noise is:

$$P_Y(\mathbf{y}|\mathbf{v}) = \frac{1}{\sqrt{(2\pi\sigma^2)^k}} \exp\left(-\frac{|\mathbf{y} - \mathbf{v}|_2^2}{2\sigma^2}\right), \tag{7}$$

in which $|\mathbf{y} - \mathbf{v}|_2^2$ denotes the L2 squared distance between $\mathbf{y}$ and $\mathbf{v}$.

Then, our Bayes optimal attack for HE-GM protocols $\mathcal{A}_{\text{HE-GM}}$ predicts the most probable input value, $\hat{v}$, given the reported $\mathbf{y}$, by following Equation (6). Remark that Equation (7) is valid for both GM and AGM as a function of their respective noise scale σ. Similar to the $\mathcal{A}_{SHE}$ attack, if the prior probability distribution $P_V(v)$ is unknown, a uniform prior may be assumed for the analysis.

4 LDP AUDITING

In this section, we introduce our LDP-Auditor framework (Section 4.1) and our distinguishability attacks considering multiple data collections (Section 4.2 and Section 4.3).

4.1 LDP-Auditor

Our LDP-Auditor framework builds upon previous work on central DP auditing [36] with slight modifications tailored for LDP auditing. This adaptation is necessary due to the intrinsic differences between the central DP and LDP models, primarily regarding the granularity of privacy and the nature of the data being protected. Figure 9 in Appendix C compares the adversarial privacy game between central and local DP. Unlike central DP, in which the adversary's objective is to distinguish between two "neighboring datasets", the LDP model shifts the focus towards distinguishing between individual "inputs". We instantiate LDP-Auditor with distinguishability attacks to construct a robust test statistic for auditing LDP mechanisms. Specifically, we can formulate a distinguishability attack as a binary hypothesis testing problem: $\mathcal{H}$: "y comes from v_1". The attacker receives an output drawn from one of the two distributions $\mathcal{M}(v_1)$ or $\mathcal{M}(v_2)$ and has to infer whether the input was v_1 or not. If the algorithm $\mathcal{M}$ is (ϵ, δ)-LDP, then no distinguishability attacks can be too accurate [38]. Specifically, for any distinguishability attack $\mathcal{A}$, we can statistically measure LDP re-writing Equation (1) as:

$$\underbrace{\Pr[\mathcal{A}(\mathcal{M}(v_1)) = v_1]}_{\text{True Positive Rate (TPR)}} \leq e^{\epsilon} \cdot \underbrace{\Pr[\mathcal{A}(\mathcal{M}(v_2)) = v_1]}_{\text{False Positive Rate (FPR)}} + \delta. \quad (8)$$

If $\mathcal{M}$ satisfies (ϵ, δ)-LDP, then $\epsilon \geq \ln\left(\frac{\text{TPR}-\delta}{\text{FPR}}\right)$. In this formulation, the TPR is the probability that the attack correctly identifies y as coming from v_1, and the FPR is the likelihood that y is incorrectly attributed to v_1 when it comes from v_2. Note that the δ term in Equation (8) reflects the privacy budget from $\mathcal{M}$ and is not an independent probability or error rate introduced by the distinguishability attack $\mathcal{A}$. However, a single run of a distinguishability attack is typically not sufficient to draw meaningful conclusions due to the inherent variability in the mechanism's outputs. Thus, to ensure the robustness of our empirical privacy loss estimation and account for statistical uncertainty, LDP-Auditor runs for multiple trials T in order to compute the TPR and FPR from Equation (8). Then, to affirm that our empirical privacy loss estimation is valid with a probability greater than $1-\alpha$, we use Clopper-Pearson confidence intervals[1] [22] to establish a lower bound $\hat{p}_1$ for the FPR and an upper bound $\hat{p}_0$ for the TPR, each with a confidence of $1-\alpha/2$. As a consequence, we can be confident that our empirical privacy loss estimation $\epsilon_{emp} = \ln\left(\frac{\hat{p}_0-\delta}{\hat{p}_1}\right)$, holds with probability $1-\alpha$. This procedure is outlined in Algorithm 1, and we prove its correctness in Theorem 1. The proof of Theorem 1 is deferred to Appendix E.

Theorem 1 (Correctness of LDP-Auditor). *Given black-box access to an LDP mechanism $\mathcal{M}$, and a distinguishability attack $\mathcal{A}$, for any two distinct values v_1, v_2, a number of trials T, and a statistical confidence α, if LDP-Auditor in Algorithm 1 returns ϵ_{emp}, then, with probability $1-\alpha$, $\mathcal{M}$ does not satisfy (ϵ', δ)-LDP for any $\epsilon' < \epsilon_{emp}$.*

[1]We briefly describe the generic Clopper-Pearson method in Appendix D.

Algorithm 1 LDP-Auditor.

Input : Theoretical ϵ and δ, LDP protocol $\mathcal{M}$, distinguishability attack $\mathcal{A}$, values $v_1, v_2 \in V$, trial count T, confidence level α.
Output : Estimated privacy loss ϵ_{emp}.

TP = 0, FP = 0 ▷ True Positive (TP) and False Positive (FP)
for $i \in [T]$ **do**
  if $\mathcal{A}(\mathcal{M}(v_1)) = v_1$ TP = TP + 1
  if $\mathcal{A}(\mathcal{M}(v_2)) = v_1$ FP = FP + 1
end for
$\hat{p}_0$ = ClopperPearsonLower(TP, T, $\alpha/2$)
$\hat{p}_1$ = ClopperPearsonUpper(FP, T, $\alpha/2$)
return : $\epsilon_{emp} = \ln((\hat{p}_0 - \delta)/\hat{p}_1)$

Accounting for statistical uncertainty. We highlight that when we refer to ϵ_{emp} as an empirical lower bound with probability $1-\alpha$, this naming is solely due to the inherent randomness of the Monte Carlo sampling process, without the need for any specific modeling or assumptions. By increasing the number of trials T, we can progressively enhance our confidence level towards 1. Furthermore, the decision to employ the Clopper-Pearson method stems from its relevance when an exact confidence interval is desired, in contrast to approximate methods (*e.g.*, heuristic approaches). This approach enables a more reliable safeguard against underestimating privacy risks, and has been widely used in central DP audit research [17, 36, 54, 60]. In this work, we utilize the Clopper-Pearson implementation provided by the `proportion_confint` method in the Python package `statsmodels` (https://pypi.org/project/statsmodels/).

Choice of parameters. Given that LDP frequency estimation protocols usually distribute noise uniformly at random, the estimation of the empirical privacy loss ϵ_{emp} is not contingent upon selecting values v_1 and v_2 to represent a "worst-case scenario", unlike in central DP audit. Considering $V = \{1, 2, \ldots, k\}$, in this work, we set $v_1 = 1$ and $v_2 = 2$. The performed tests revealed no statistical difference when experiments were conducted with $v_1 = 1$ and a dynamic $v_2 = \text{Uniform}(2, k)$. Considering the experimental setup parameters, the number of trials T and the confidence level $1-\alpha$ should be chosen to balance computational efficiency with the robustness of the empirical privacy loss estimation. Typically, a larger T enhances the reliability of ϵ_{emp} estimates, while a smaller α increases the confidence in these estimates. In this work, we recommend selecting T to be sufficiently large to ensure stable estimates across multiple experiments (*e.g.*, we set $T = 10^6$) and setting α to reflect a high confidence level, such as 0.05 or 0.01, to underpin the statistical significance of the empirical findings.

Limits on the empirical privacy loss estimation. The ϵ_{emp} reported by Algorithm 1 is upper bounded by the theoretical ϵ but also by an upper bound imposed by Monte Carlo estimation, which will be denoted by ϵ_{OPT} and depends on α and T. For instance, let $\alpha = 0.01$ to get a 99%-confidence bound and $T = 10^4$ trials. Even if we get perfect inference accuracy with TP = T and FP = 0, the Clopper-Pearson confidence interval would produce $\hat{p}_0 = 0.9994$ and $\hat{p}_1 = 0.0006$, which implies an empirical privacy loss of $\epsilon_{emp} = 7.42$. This means, with 99% probability, the true ϵ is at least 7.42, and $\epsilon_{OPT}(\alpha, T) = 7.42$.

4.2 LDP-Auditor for Longitudinal Studies

In practice, the server often needs to collect users' data periodically throughout multiple data collections (*i.e.*, *longitudinal studies*). Nevertheless, in the worst-case, one known result in (L)DP is that **repeated data collections have a linear privacy loss due to the sequential composition** [30]. This occurs because attackers can exploit "averaging attacks" to distinguish the user's actual value from the added noise. For this reason, well-known LDP mechanisms for longitudinal studies such as RAPPOR [32] (deployed in Google Chrome) and *d*BitFlipPM [26] (deployed in Windows 10), were designed with a *memoization-based* solution. We discuss how to audit LDP mechanisms based on memoization in Appendix F.

Given τ data collections, we aim to audit the empirical privacy loss of LDP protocols in comparison to the upper bound $\tau\epsilon$-LDP imposed by the (L)DP sequential composition. Our main motivation is to evaluate how tight the sequential composition is for LDP protocols. Furthermore, this audit will provide insights into the privacy implications of real-world applications similar to those implemented by Apple [59], in which memoization was not employed.

In Algorithm 2, we present the extension of distinguishability attacks on LDP protocols to longitudinal studies $\mathcal{A}^L$. In this context, the adversary's objective remains the same: to predict the user's true value by determining the most probable value that would have generated the reported value y^t after τ data collections. Notably, the adversary now possesses an increased knowledge due to random fresh noise being added to the user's value v over τ times. To perform the "averaging attack", in each data collection, the adversary constructs the "support set" based on the reported value y^t and LDP mechanism $\mathcal{M}$. The support set is then used to increment the knowledge (*i.e.*, count) about the user's true value and what constitutes noisy data, ultimately predicting $\hat{v}$. We highlight that the exceptions are HE-based protocols in which the notion of a support set is not applicable, namely SHE, GM and AGM, rendering Algorithm 2 inapplicable. In these protocols, Laplace or Gaussian noise with a mean of 0 is added in each data collection. Consequently, the "averaging attack" is straightforward as it involves determining $\hat{v}$ by taking the `argmax` of the summation of all reports. Formally, this is expressed as $\hat{v} = \text{argmax}\left(\sum_{t=1}^{\tau} \mathbf{y}^t\right)$.

Finally, our LDP-Auditor framework (Algorithm 1) can be used to estimate the privacy loss of LDP protocols in longitudinal studies. To achieve this, one can simply replace "$\mathcal{A}(\mathcal{M}(v))$" in Lines 3 and 4 of Algorithm 1 with "$\mathcal{A}^L(v)$", *i.e.*, the distinguishability attack outlined in Algorithm 2, which already takes into account $\mathcal{M}$.

Algorithm 2 Distinguishability Attack in Longitudinal Study: $\mathcal{A}^L$.

Input : User value v, privacy guarantee ϵ, LDP protocol $\mathcal{M}$, number of data collections τ.
Output : Predicted value $\hat{v}$.

Initialize a k sized zero-vector $\mathbf{z} = [0, 0, \ldots, 0]$
for $t \in [\tau]$ **do:**
  User-side randomization $y^t = \mathcal{M}(v)$
  Given y^t, adversary construct support set $\mathbb{1}_{\mathcal{M}}$
  for $v \in \mathbb{1}_{\mathcal{M}}$ **do:**
    Increment count $\mathbf{z}[v] = \mathbf{z}[v] + 1$
  end for
end for
Predict $\hat{v} = \text{argmax}(\mathbf{z})$
return : $\hat{v}$

4.3 LDP-Auditor for Multidimensional Data

Another dimension of interest to the server is *multidimensional data* (*i.e.*, $d \geq 2$ attributes), aiming to enable more comprehensive decision-making. Considering potential correlations among these attributes, the principles of DP sequential composition [30] remain applicable in this context. Therefore, the existing solutions for multidimensional data, represented as $\mathbf{v} = [v_1, v_2, \ldots, v_d]$, include:

- **Splitting (SPL):** This naïve method involves partitioning the privacy budget ϵ among the d attributes, collecting each attribute under $\frac{\epsilon}{d}$-LDP. Examples based on this SPL solution are the LoPub [57] and Castell [41] mechanisms, which are designed for joint distribution estimation.
- **Sampling (SMP):** In this approach, users are divided into d disjoint sub-groups. Each sub-group $j \in [d]$ then reports the j-th attribute under ϵ-LDP. Example of mechanisms using the SMP solution include CALM [75] and FELIP [24], proposed for marginal estimation, and [55, 64, 69], which introduced LDP mechanisms for mean estimation.
- **Random Sampling Plus Fake Data (RS+FD) [5]:** In this solution, each user samples a single attribute $j \in [d]$ to report v_j under ϵ'-LDP and reports uniform fake data for the $d-1$ non-sampled attributes. Because the sampling result is not disclosed to the aggregator, there is amplification by sampling [13, 43]. For this reason, RS+FD utilizes an amplified privacy budget $\epsilon' = \ln\left(d \cdot (e^{\epsilon} - 1) + 1\right)$ for the sampled attribute. An example based on RS+FD is the GRR-FS mechanism [16], designed for node-level LDP on graph data, to enable training of graph neural networks.

Upon closer examination of the three solutions, one can notice that both SPL and SMP solutions can be considered as straightforward instances of reporting one attribute with a given LDP mechanism (one at a time for SPL). Consequently, our LDP-Auditor framework can be directly used to estimate empirical privacy losses ϵ_{emp} for LDP mechanisms following the SPL and SMP solutions. Therefore, in this work, our focus shifts towards auditing the RS+FD solution, for which there is a privacy amplification effect due to uncertainty on the server side.

In Algorithm 3, we introduce the distinguishability attack designed for LDP protocols following the RS+FD solution, denoted as $\mathcal{A}^{\text{RS+FD}}$. Here, the adversary's objective is twofold: first, to predict the attribute that the user has sampled, and subsequently, to predict the user's actual value. Since each user selects an attribute $j \in [d]$ uniformly at random, the Bayes optimal guess for the adversary is $\hat{j} = \text{Uniform}([d])$. Once the attribute is predicted, the adversary constructs the "support set" based on the reported value $y_{\hat{j}}$ and LDP mechanism $\mathcal{M}$. With the support set, as in Section 3, the adversary predicts the user's value $\hat{v}_{\hat{j}}$.

Finally, we extend our LDP-Auditor framework for RS+FD protocols in Algorithm 4. The main change is due to the multidimensional data setting, for which we define $\mathbf{v_1}$ and $\mathbf{v_2}$ in Lines 1 and 2 of Algorithm 4. The test statistic remains unchanged, as it is derived from distinguishability attacks as per Algorithm 3. Notice that one

main difference with RS+FD auditing is that even if the user did not sample the attribute $\hat{j}$, the attack can still predict the user's value v_j correctly due to uniform fake data generation for that attribute. Our goal is thus to audit if RS+FD satisfies the claimed ϵ-LDP guarantee with amplification by sampling. Algorithm 4 builds upon the foundational principles established in Section 4.1 and in Algorithm 1 and, consequently, the framework's correctness and reliability extend to this adaptation as well.

Algorithm 3 Distinguishability Attack on RS+FD: $\mathcal{A}^{\text{RS+FD}}$.

Input : User values $\mathbf{v} = [v_1, v_2, \ldots, v_d]$, privacy guarantee ϵ, RS+FD protocol $\mathcal{M}$.
Output : Predicted value $\hat{v}_{\hat{j}}$.

for $i \in [d]$ **do:** ▷ *cf.* RS+FD [5, Algorithm 1]
  User-side randomization $y_i = \mathcal{M}(v_i)$
end for
Adversary predict user's sampled attribute $\hat{j} = \text{Uniform}([d])$
Given $y_{\hat{j}}$, construct support set $\mathbb{1}_{\mathcal{M}}$
Predict $\hat{v}_{\hat{j}} = \text{Uniform}(\mathbb{1}_{\mathcal{M}})$ ▷ *cf.* Section 3
return : $\hat{v}_{\hat{j}}$

Algorithm 4 LDP-Auditor for RS+FD Protocols.

Input : Theoretical ϵ, LDP protocol $\mathcal{M}$, distinguishability attack $\mathcal{A}^{\text{RS+FD}}$, values $v_1, v_2 \in V$, trial count T, confidence level α.
Output : Estimated privacy loss ϵ_{emp}.

$\mathbf{v_1} = [v_1, v_1, \ldots, v_1]_{1\times d}$, $\mathbf{v_2} = [v_2, v_2, \ldots, v_2]_{1\times d}$
TP = 0, FP = 0 ▷ True Positive (TP) and False Positive (FP)
for $i \in [T]$ **do**
  if $\mathcal{A}^{\text{RS+FD}}(\mathbf{v_1}) = v_1$ TP = TP + 1
  if $\mathcal{A}^{\text{RS+FD}}(\mathbf{v_2}) = v_1$ FP = FP + 1
end for
$\hat{p}_0 = \text{ClopperPearsonLower}(\text{TP}, T, \alpha/2)$
$\hat{p}_1 = \text{ClopperPearsonUpper}(\text{FP}, T, \alpha/2)$
return : $\epsilon_{emp} = \ln(\hat{p}_0/\hat{p}_1)$

5 EXPERIMENTAL EVALUATION

This section presents our experimental setting to assess the proposed audit framework as well as the main results obtained.

5.1 General Setup of Experiments

For all experiments, we have used the following setting:

- **Environment.** All algorithms are implemented in Python 3 with the Numpy [62], Numba [42], Ray [51], Multi-Freq-LDPy [8] and pure-LDP [23, 46] libraries, and run on a local machine with 2.50GHz Intel Core i9 and 64GB RAM. Our LDP-Auditor tool is open-sourced in a GitHub repository [4].
- **Audit parameters.** We set $T = 10^6$ trial counts and use Clopper-Pearson confidence intervals with $\alpha = 0.01$ (*i.e.*, our estimates hold with 99% confidence). These parameters establish the Monte Carlo upper bound as $\epsilon_{OPT} = 12.025$.
- **Stability.** Since LDP protocols are randomized, we report average results with standard deviation over 5 runs.

5.2 Main Auditing Results

We begin by presenting our main LDP auditing results, considering:

- **LDP protocols.** We audit the eight ϵ-LDP frequency estimation protocols described in Section 3.1 and the six (ϵ, δ)-LDP frequency estimation protocols described in Section 3.2.
- **Theoretical upper bound.** We evaluated the LDP frequency estimation protocols in high, mid and low privacy regimes over the range $\epsilon \in \{0.25, 0.5, 0.75, 1, 2, 4, 6, 10\}$. The chosen range for ϵ follows the state-of-the-art LDP literature (*e.g.*, see [2, 49, 68, 72, 75]) and real-world implementations [25] (*e.g.*, RAPPOR [32] with $\epsilon = 0.5$).
- **Delta parameter.** For approximate LDP, we set $\delta = 1e^{-5}$.
- **Domain size.** We also varied the domain size $k \in \{25, 50, 100, 150, 200\}$ as it influences the performance of the distinguishability attacks.

Figure 2 illustrates the theoretical ϵ values (x-axis) versus the estimated ϵ_{emp} values (y-axis), demonstrating the comparison across various domain sizes k, for the eight ϵ-LDP frequency estimation protocols: GRR, SS, SUE, OUE, BLH, OLH, SHE and THE. Similarly, Figure 3 presents analogous plots for the six (ϵ, δ)-LDP frequency estimation protocols: AGRR, ASUE, ABLH, AOLH, GM and AGM. *Henceforth, when discussing our results, the notation "(A)GRR" will be used whenever the findings are applicable to both GRR and AGRR protocols (analogously for other LDP protocols).*

Effect of Encoding and Perturbation Functions. It is important to note that LDP frequency estimation protocols employ different encoding and perturbation functions, leading to varying levels of susceptibility to distinguishability attacks [9, 31]. Notably, as shown in Figure 2 and Figure 3, one can notice that *(A)GRR is the unique LDP protocol that achieves tight empirical privacy estimates for ϵ_{emp}*. As described in Section 3.1, auditing (A)GRR's privacy guarantees is straightforward since there is no specific encoding (*i.e.*, the input and output spaces are equal). Conversely, all other LDP protocols (*i.e.*, SS, UE-, LH- and HE-based) incorporate specific pre-processing encoding functions, which may result in information loss and/or additional randomness.

For instance, (A)BLH hashes the input set V of size k to $\{0, 1\}$ and, thus results in excessive loss of information due to collisions. Even if the bit is transmitted correctly after the (A)GRR perturbation, the server can only obtain one bit of information about the input (*i.e.*, to which half of the input domain the value belongs to). For these reasons, *BLH consistently led to the worst auditing results* among the ϵ-LDP protocols with a "flat" $\epsilon_{emp} < 1$ estimation after $\epsilon \geq 2$. Indeed, although both (A)LH protocols present similar empirical privacy losses ϵ_{emp} in high privacy regimes (*the lowest among all other LDP protocols*), the difference is remarkable in favor of (A)OLH in mid to low privacy regimes. Thus, *(A)OLH preserves more utility than (A)BLH, while providing tighter privacy loss estimation.*

Concerning the SS protocol that reports a subset Ω of ω values, one can note from Figure 2 that the empirical privacy loss ϵ_{emp} demonstrated similar results to other LDP protocols in high privacy regimes. However, an exception occurs in low privacy regimes, in which SS equals GRR due to a subset size $\omega = 1$, resulting in tight estimates for ϵ_{emp}. Regarding UE-based protocols, in high-privacy regimes ($\epsilon \leq 1$), both SUE and OUE presented similar empirical privacy estimates for ϵ_{emp} in Figure 2. In mid-privacy regimes

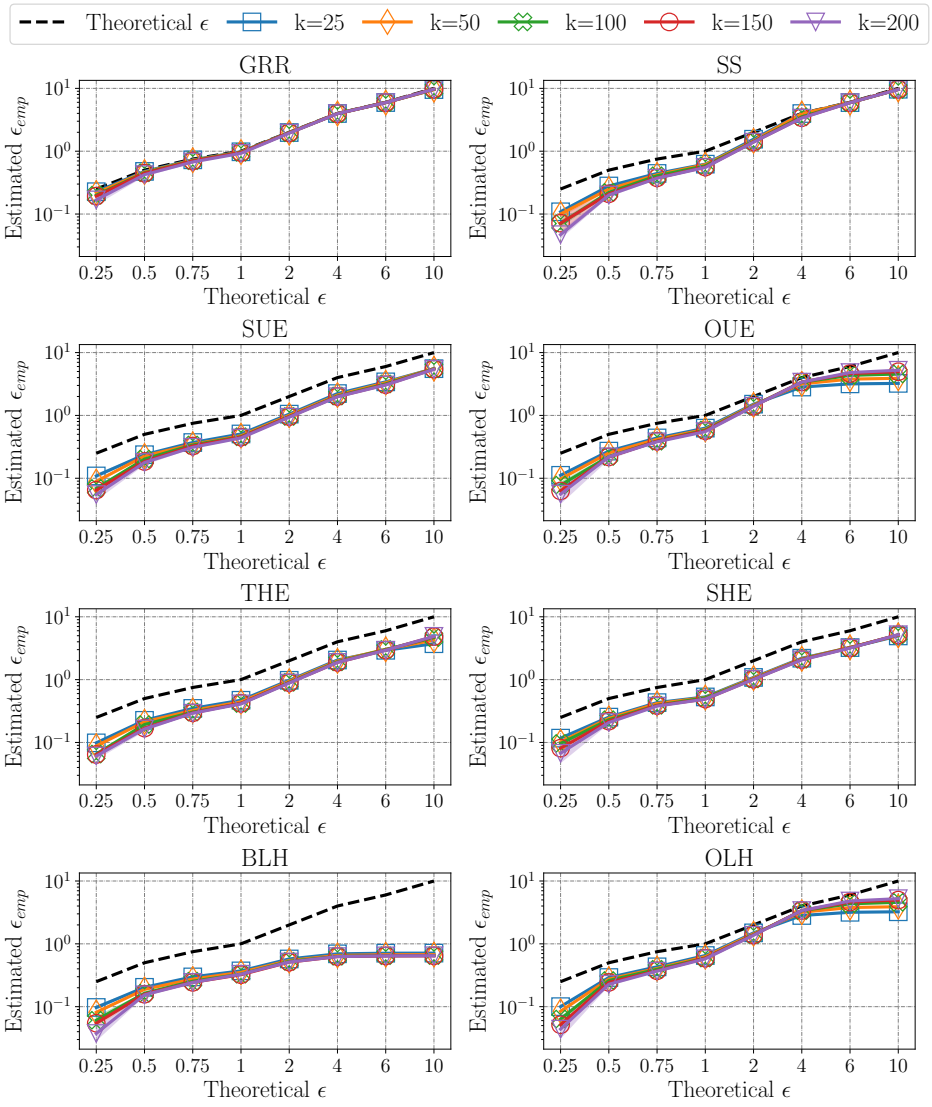


Figure 2: Theoretical ϵ values (x-axis) versus estimated ϵ_{emp} values (y-axis) using our LDP-Auditor framework with $\delta = 0$. We compare different domain sizes k for eight state-of-the-art ϵ-LDP frequency estimation protocols: GRR [37], SS [65, 73], SUE [32], OUE [66], BLH [15], OLH [66], SHE [29] and THE [66].

($1 < \epsilon \leq 4$), OUE presented higher empirical privacy losses ϵ_{emp} than SUE. However, OUE reached a "plateau" estimation for ϵ_{emp} in low privacy regimes ($\epsilon > 4$), explained by an upper bound on the distinguishability attack (see [31]). This plateau behavior is also observed for the (A)OLH protocol in low privacy regimes due to a comparable upper bound on the attacker effectiveness. Comparing approximate- and pure-SUE protocols, similar results were noticed for (A)SUE in Figure 2 and Figure 3, considering all privacy regimes.

Lastly, for HE-based protocols, similar estimates for ϵ_{emp} were observed across all privacy regimes for both ϵ-LDP protocols, namely SHE and THE, in Figure 2, albeit with varying sensitivity to the domain size k (discussed afterwards). In contrast, from Figure 3, one can notice that the (ϵ, δ)-LDP GM protocol led to the worst auditing results among all LDP protocols. Therefore, *in addition to AGM's ability to preserve greater utility than GM, it also offers more precise empirical privacy loss estimations.*

Impact of domain size. As the domain size k increases, one can observe in Figure 2 and Figure 3 a direct impact on the empirical privacy loss estimation of ϵ_{emp} for all LDP protocols, in which the gap with the theoretical ϵ increases. However, the impact is minor for the (A)GRR protocol, even in high privacy regimes. Conversely, for all other LDP protocols, this impact is substantial, with empirical ϵ_{emp} estimates ranging within $\leq$ 2.5x of the theoretical ϵ (when $k = 25$) up to $\leq$ 5x (when $k = 200$). These results are consistent with the distinguishability attack effectiveness, which decreases according to higher k (*i.e.*, more uncertainty) [9, 31]. For instance,

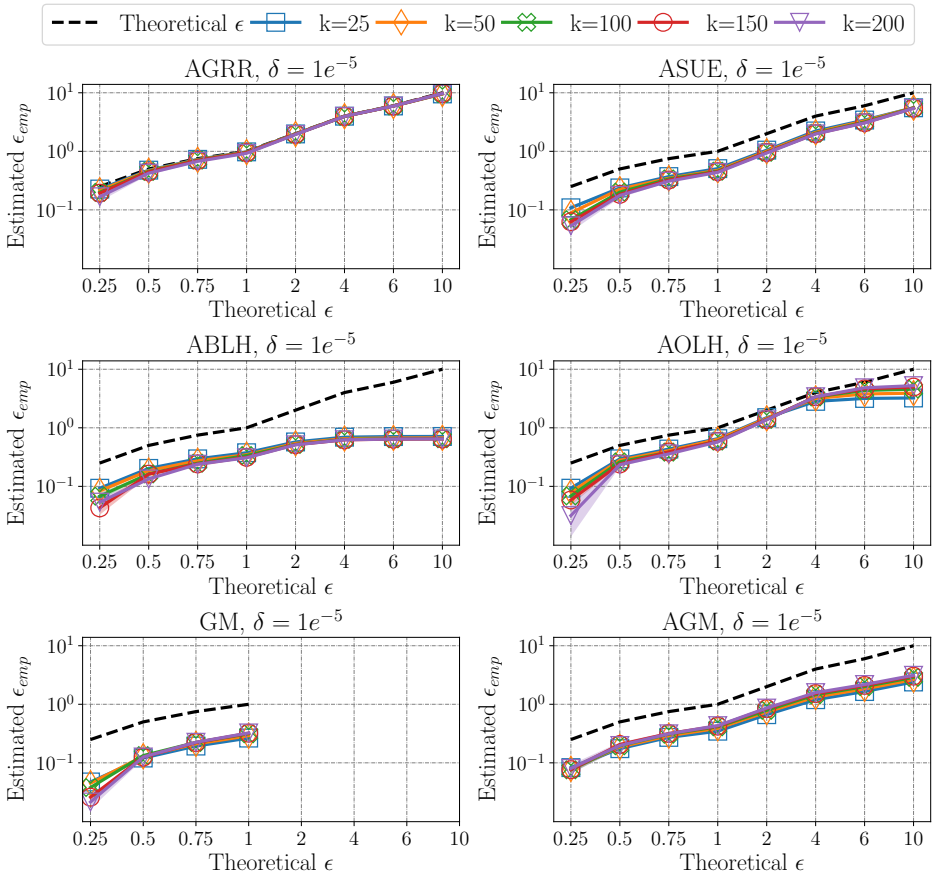


Figure 3: Theoretical ϵ values (x-axis) versus estimated ϵ_{emp} values (y-axis) using our LDP-Auditor framework with $\delta = 1e^{-5}$. We compare different domain sizes k for six state-of-the-art (ϵ, δ)-LDP frequency estimation protocols: AGRR [69], ASUE [69], ABLH [69], AOLH [69], GM [30] and AGM [14]. For GM, we only audit for certifiable theoretical upper bounds $\epsilon \leq 1$.

in the case of GRR, the probability $p = \frac{e^{\epsilon}}{e^{\epsilon}+k-1}$ of being "honest" in Equation (2) decreases proportionally to k. In other mechanisms, there is a higher likelihood of introducing noise in the output y, such as by flipping more bits from 0 to 1 in (A)UE protocols.

Nevertheless, exceptions exist for both OUE and OLH protocols, in which in low privacy regimes (when $\epsilon \geq 4$), a larger domain size k leads to tighter estimates of ϵ_{emp} than smaller domain sizes. Although to a small extent, the THE protocol also yields more accurate estimates for higher k when $\epsilon = 10$. Taking OUE as an example, these results can be attributed to the fact that the bit corresponding to the user's value is transmitted with a random probability of $\frac{1}{2}$ (*cf.* Equation (3)). Consequently, if the domain size is small, it results in a higher false positive rate, which subsequently decreases the estimated empirical privacy loss ϵ_{emp}.

Generality of Our Findings. Overall, the gap between empirical ϵ_{emp} and theoretical ϵ privacy guarantees tends to widen in high privacy regimes (*i.e.*, lower ϵ values). This trend is particularly pronounced when considering the sensitivity of different LDP protocols to the domain size. Lastly, we highlight that all ϵ-LDP and (ϵ, δ)-LDP frequency estimation protocols audited herein are building blocks of LDP mechanisms for more complex tasks such as: heavy hitter estimation [15, 68], joint distribution estimation [24, 41, 57, 75], frequent item-set mining [67, 72], machine learning [49, 74], frequency estimation of multidimensional data [5, 55, 64] and frequency monitoring [7, 10, 26, 32, 63]. Thus, our audit results provide generic insights that shed light on several critical factors influencing the estimation of the local privacy loss.

5.3 Case Study #1: Approximate- *VS* Pure-LDP

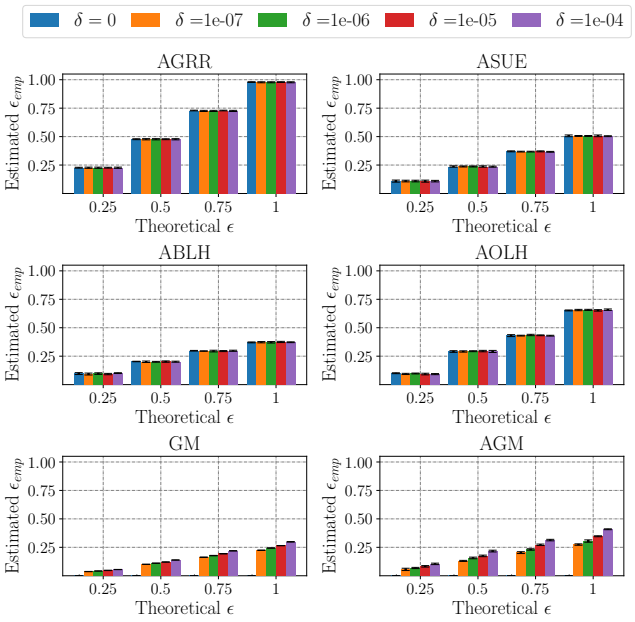
In theory, Bun et al. [18] proved that in the local DP model, approximate privacy is actually never more useful than pure privacy. We will now compare approximate- and pure-LDP by assessing the impact of δ on the LDP auditing process. In these experiments, we use the following parameter values:

- **LDP protocols.** We audit the six (ϵ, δ)-LDP protocols described in Section 3.2.
- **Theoretical upper bound.** Because GM requires $\epsilon \leq 1$ [30], we vary the privacy guarantee only in high privacy regimes, within the range $\epsilon \in \{0.25, 0.5, 0.75, 1\}$.
- **Delta parameter.** We vary the δ parameter within the range $\delta \in \{0, 1e^{-7}, 1e^{-6}, 1e^{-5}, 1e^{-4}\}$; $\delta = 0$ means ϵ-LDP.

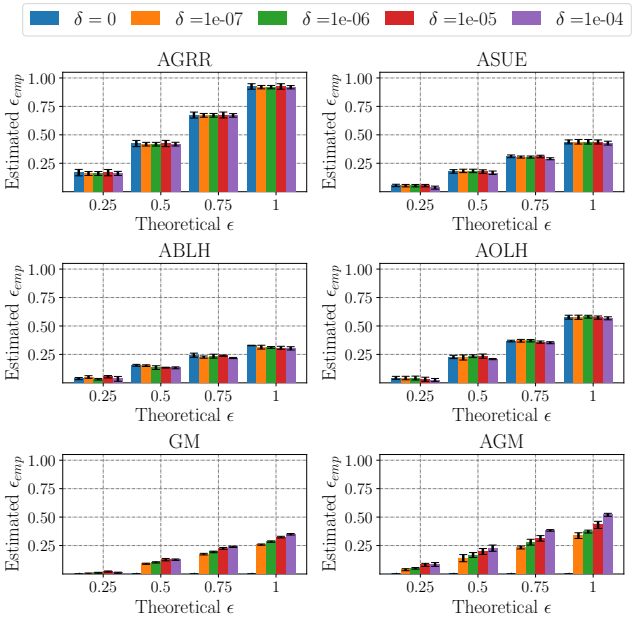
- **Domain size.** We vary the domain size $k \in \{25, 100, 150, 200\}$. We present results for $k \in \{25, 200\}$ in the main paper and defer the others to Appendix G.1

Figure 4 illustrates the theoretical ϵ values (x-axis) versus estimated ϵ_{emp} values (y-axis) when varying the δ parameter and domain size $k \in \{25, 200\}$, using our LDP-Auditor framework. Note that for both GM and AGM protocols, there is no ϵ_{emp} value when $\delta = 0$, as these protocols do not have pure ϵ-LDP variations.

Interestingly, for protocols such as AGRR, ASUE, ABLH and AOLH, our observations corroborate the theoretical assertions made by Bun et al. [18] regarding the comparative utility of approximate versus pure privacy. More precisely, Figure 4 and Figure 10 (for $k \in \{100, 150\}$) indicate that variations in δ do not significantly alter the estimated privacy loss ϵ_{emp} across these protocols. This consistency in ϵ_{emp} values, irrespective of δ adjustments, suggests that for LDP protocols with a finite range, the audit outcomes for approximate-privacy closely align with those for pure-privacy. Conversely, the GM and AGM protocols exhibit distinct behaviours. More precisely, as δ increases, signaling a relaxation in the privacy constraint, we observe a narrowing gap between theoretical ϵ and empirical ϵ_{emp} values. This trend highlights a crucial aspect of LDP protocols with an infinite range, in which allowing for a nonzero δ directly influences the perceived privacy protection, leading to a more pronounced estimation of the privacy loss. Finally, the impact of the domain size on the estimated privacy loss has a minor effect on the AGRR, ASUE, ABLH and AOLH protocols, with a decreasing ϵ_{emp} value for higher k. In contrast, for both GM and AGM protocols, the estimated ϵ_{emp} values increase (*i.e.*, indicating less privacy) as k increases.



(a) Domain size $k = 25$.



(b) Domain size $k = 200$.

Figure 4: Theoretical ϵ values (x-axis) versus estimated ϵ_{emp} values (y-axis) using our LDP-Auditor framework. We assess different privacy guarantees for six (ϵ, δ)-LDP protocols across domain sizes $k \in \{25, 200\}$. The special case $\delta = 0$ corresponds to pure ϵ-LDP, for which GM and AGM do not satisfy.

5.4 Case Study #2: Auditing the Privacy Loss of Local Hashing Encoding Without LDP

As discussed previously in Section 5.2, both LH protocols present the least tight estimates for ϵ_{emp} in high privacy regimes. Even worse, BLH's estimated privacy loss remains below $\epsilon_{emp} < 1$ for $\epsilon \geq 2$, leading to empirical privacy losses $\leq$ 10x of the theoretical ϵ. Motivated by these observations, we performed an additional study **to audit the impact of local hashing encoding but with no LDP perturbation** (*i.e.*, $\epsilon = +\infty$), which we refer to as Local Hashing Only (LHO). More precisely, the LHO reporting mechanism is $\text{LHO}(v) := \langle \text{H}, \text{H}(v) \rangle$, and we used the same distinguishability attack $\mathcal{A}_{\text{LH}}$ described in Section 3 to attack LHO. For these experiments, we use the following parameter values:

- **LHO hash domain.** We vary the hash domain $[g]$ within the range $g \in \{2, 4, 6, 8, 10\}$.
- **Domain size.** We vary the domain size within the range $k \in \{25, 50, 100, 150, 200\}$.

Figure 5 presents the estimated ϵ_{emp} values (x-axis) for LHO protocols according to the hash domain sizes g (y-axis) using our LDP-Auditor framework for different domain sizes k. Observations from Figure 5 underscore that, even for a binary hash domain ($g = 2$), the estimated privacy loss remains $\epsilon_{emp} < 1$, aligning with high privacy regimes suitable for real-world applications. Indeed, even though there is no LDP randomization of the hashed value $h \in \{0, 1\}$, the adversary still has a random guess on the support set $\mathbb{1}_{\text{LH}}$. Given a general (universal) family of hash functions $\mathscr{H}$, each input value $v \in V$ is hashed into a value in $[g]$ by a hash function $\text{H} \in \mathscr{H}$, and the universal property requires

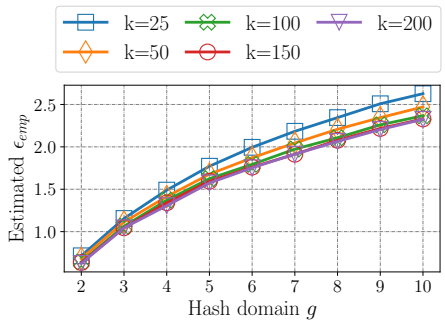


Figure 5: Estimated ϵ_{emp} (y-axis) versus hash domain g (x-axis) using our LDP-Auditor framework comparing different domain sizes k for LH encoding with no LDP randomization.

$\forall v_1, v_2 \in V, v_1 \neq v_2 : \Pr_{H \in \mathcal{H}}[H(v_1) = H(v_2)] \leq \frac{1}{g}$. In other words, approximately k/g values can be mapped to the same hashed value $h = H(v)$ in $[g]$. Although local hashing pre-processing by itself has no proven DP guarantees, this significant loss of information in the encoding step suggests potential privacy gains for LH protocols due to the presence of many random collisions. In a similar context, DP-Sniper [17], a method developed to finds violations of DP, also encountered difficulties estimating ϵ for the original RAPPOR [32], which is based on Bloom filters and employs hash functions.

One could expect a similar privacy gain for other LDP mechanisms based on sketching such as Apple's Count-Mean Sketch (CMS) [59] and Hadamard [2] mechanisms, which we leave as for future audit investigations. Furthermore, as we increase the hash domain size $g > 2$ without introducing any LDP perturbation, the estimated ϵ_{emp} starts to rise, achieving medium privacy regimes $1 < \epsilon_{emp} \leq 2.5$. This outcome is expected since preserving more information during the encoding step decreases the support set size $|\mathbb{1}_{\text{LH}}|$, which naturally enhances the accuracy of the distinguishability attack $\mathcal{A}_{\text{LH}}$. Therefore, the estimated privacy loss ϵ_{emp} for LH-based protocols will be lower if the domain size k is high and/or if the new hashed domain g is small.

5.5 Case Study #3: Auditing the LDP Sequential Composition in Longitudinal Studies

As discussed in Section 4.2, we aim to audit the empirical privacy loss of LDP protocols in longitudinal studies (*i.e.*, τ data collections). This will allow to assess the gap between empirical local privacy loss estimation and the theoretical upper bound imposed by the (L)DP sequential composition. For these experiments, we use both Algorithms 1 and 2 with the following parameter values:

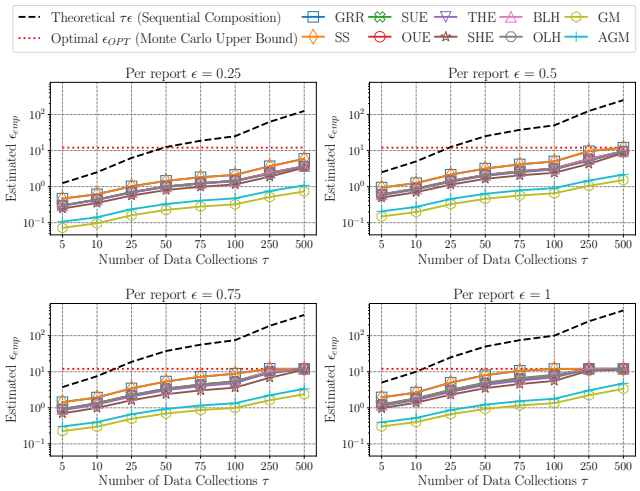
- **LDP protocols.** We audit the eight ϵ-LDP protocols from Section 3.1. Additionally, in light of the findings presented in Section 5.3, we only audit two (ϵ, δ)-LDP protocols that exhibit sensitivity to δ; namely, GM and AGM.
- **Number of data collections.** We vary the number of data collections in the range $\tau \in \{5, 10, 25, 50, 75, 100, 250, 500\}$.
- **Theoretical upper bound.** We vary the per-report privacy guarantee in high privacy regimes, in the range $\epsilon \in \{0.25, 0.5, 0.75, 1\}$. By the sequential composition, the theoretical upper bound after τ data collections is $\tau\epsilon$-LDP.
- **Delta parameter.** For approximate LDP, we set $\delta = 1e^{-5}$.
- **Domain size.** We vary the domain size $k \in \{2, 25, 50, 100\}$. We present results for $k \in \{2, 100\}$ in the main paper and defer the others to Appendix G.2.

Figure 6 illustrates the estimated ϵ_{emp} values (y-axis) for the eight ϵ-LDP and both GM and AGM (ϵ, δ)-LDP protocols according to the the number of data collections τ (x-axis), per report ϵ and domain size $k \in \{2, 100\}$, using our LDP-Auditor framework. From Figure 6a, one can notice that both GRR and SS protocols have equal ϵ_{emp} estimates, as for $k = 2$, the subset size $\omega = 1$ (*i.e.*, GRR). These two LDP protocols exhibited the tightest empirical privacy estimates for ϵ_{emp}, aligning with the observations made in Section 5.2 (see Figure 8). In contrast, the approximate LDP protocols, notably GM and AGM, showed less favorable estimates for privacy loss, which corroborates the findings illustrated in Figure 3 in Section 5.2. The remaining pure-LDP protocols – SUE, OUE, BLH and OLH – display intermediate privacy loss estimates.

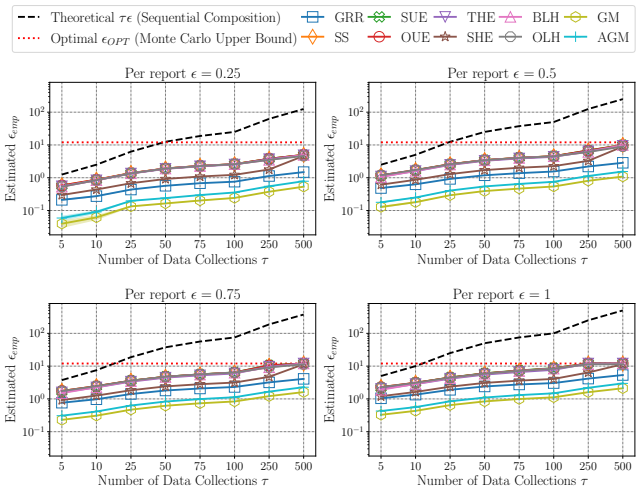
Furthermore, Figure 6b reveals that, for a larger domain size of $k = 100$, the results obtained are reversed. Among pure-LDP protocols, GRR yields the lowest ϵ_{emp} estimation for all experimented τ values, followed by the SHE protocol. The reason for this is that the probability of being "honest" $p = \frac{e^\epsilon}{e^\epsilon + k - 1}$ in Equation (2), is directly proportional to the domain size k. Therefore, even after many data collections τ, the adversary has still too much noisy data to filter, which makes the distinguishability attack less efficient. Similar to Figure 6, in Figure 11, approximate-LDP protocols (GM and AGM) led to the lowest empirical privacy loss estimates for ϵ_{emp}.

Moreover, from both Figure 6 and Figure 11, it is evident that even after $\tau = 500$, none of the LDP protocols, achieves the optimal upper bound ϵ_{OPT} imposed by the Monte Carlo estimation when the per-report privacy guarantee is too small (*i.e.*, $\epsilon = 0.25$). However, as the number of data collections becomes sufficiently large (*i.e.*, $\tau \geq 250$) and the privacy guarantee per report also increases (*e.g.*, $\epsilon \geq 0.75$), all pure-LDP protocols, with the exception of GRR, manage to achieve the Monte Carlo upper bound, resulting in $\epsilon_{emp} = \epsilon_{OPT}$. Yet, as the number of data collections becomes sufficiently large (*i.e.*, $\tau \to \infty$), we anticipate that ϵ_{emp} will converge to ϵ_{OPT} for all LDP protocols even when the per-report $\epsilon < 0.5$.

These results are quite surprising since one would imagine the privacy leakage to be higher for repeated data collections when random fresh noise is added per report. Nevertheless, as the domain size increases, the performance of the distinguishability attack decreases [9, 31]. As a consequence, for real-world deployments with substantial domain sizes (*e.g.*, list of Internet domains), exclusively relying on theoretical ϵ-LDP guarantees may prove unrealistic. Privacy auditing becomes imperative in such scenarios, to establish appropriate privacy parameters, thus avoiding adding more noise than required. *Notably, these auditing results emphasize a crucial aspect for longitudinal studies: a substantial gap exists between theory (sequential composition) and practice (LDP auditing).* To narrow this gap, one could consider designing more powerful attacks for longitudinal studies beyond those proposed here in Algorithm 2. Alternatively, research efforts could be directed towards developing more sophisticated compositions for ϵ-LDP mechanisms.



(a) Domain size $k = 2$.



(b) Domain size $k = 100$.

Figure 6: Estimated ϵ_{emp} (y-axis) versus the number of data collections τ (x-axis) using our LDP-Auditor framework for different domain sizes $k \in \{2, 100\}$. We vary the per report ϵ-LDP guarantee for the following LDP frequency estimation protocols: GRR, SS, SUE, OUE, BLH, OLH, SHE, THE, GM and AGM. For both approximate-LDP protocols, namely GM and AGM, $\delta = 1e^{-5}$.

5.6 Case Study #4: LDP Auditing with Multidimensional Data

As discussed in Section 4.3, our audit results outlined in Section 5.2 are also valid for LDP mechanisms based on the standard SPL and SMP solutions for multidimensional data. Thus, in this section, we aim to audit LDP protocols following the RS+FD [5] solution. For these experiments, we use both Algorithms 3 and 4, considering:

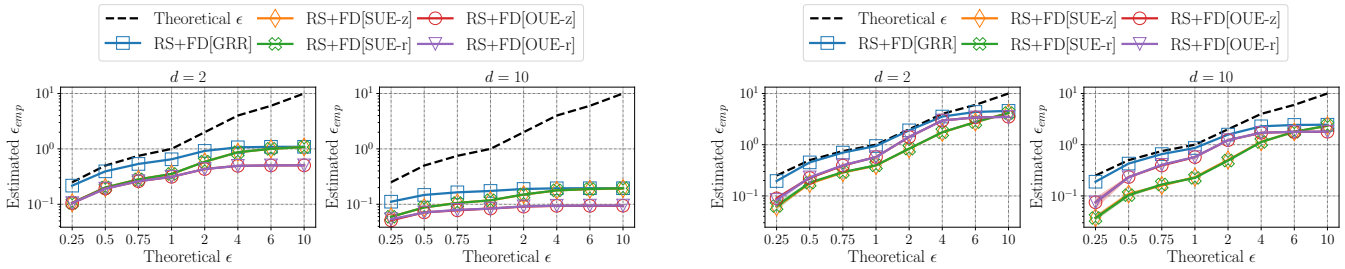
- **LDP protocols.** We audit five ϵ-LDP RS+FD protocols: RS+FD[GRR], RS+FD[SUE-z], RS+FD[SUE-r], RS+FD[OUE-z] and RS+FD[OUE-r]. The difference between UE-z and UE-r lies on how to generate the fake data [5]. More precisely, UE-z initializes a zero-vector and UE-r initializes a random one-hot-encoded vector. Next, SUE or OUE is used to sanitize these vectors.
- **Theoretical upper bound.** We vary the theoretical privacy parameter ϵ in high, mid and low privacy regimes over the same range $\epsilon \in \{0.25, 0.5, 0.75, 1, 2, 4, 6, 10\}$ as in Section 5.2.
- **Domain size and number of attributes.** We vary the domain size as $k \in \{2, 25, 50, 100\}$ and we vary the number of attributes over $d \in \{2, 10\}$. When $d = 2$, $\mathbf{k} = [2, 2]$, $\mathbf{k} = [25, 25]$, $\mathbf{k} = [50, 50]$ and $\mathbf{k} = [100, 100]$ and, in a similar way for $d = 10$. We present results for $k \in \{2, 100\}$ in the main paper and defer the others to Appendix G.3.

Figure 7 illustrates the comparison of theoretical ϵ values (x-axis) with estimated ϵ_{emp} values (y-axis) for the five RS+FD protocols, based on the number of attributes d and domain size $k \in \{2, 100\}$, utilizing our LDP-Auditor framework. From Figure 7, it is clear that, once again, GRR exhibits tighter empirical privacy losses ϵ_{emp} than UE-based protocols following the RS+FD solution. However, in contrast to Section 5.2, the estimated ϵ_{emp} for GRR now displays a "plateau behaviour" after theoretical $\epsilon \geq 4$. This plateau arises because the probability of reporting the true value under GRR reaches high values with $\epsilon \geq 4$. Notably, among the family of UE protocols, SUE demonstrates a tighter empirical ϵ_{emp} than OUE when the domain is binary (see Figure 7a). However, SUE exhibits lower ϵ_{emp} than OUE when $k = 100$ (see Figure 7b). This observation can be attributed to the advantage of SUE in transmitting the true bit with a probability $p > \frac{1}{2}$, while OUE has $p = \frac{1}{2}$. Consequently, the distinguishability attack achieves higher accuracy for SUE, increasing the true positive rate and decreasing the false positive rate, resulting in higher ϵ_{emp} estimates. Moreover, different fake data generation procedures for UE protocols (UE-z *vs* UE-r) did not result in significant changes in the audit results.

Another intriguing result is that the empirical privacy loss is lower for a binary domain compared to when $k = 100$. This behavior is primarily due to the impact of fake data on distinguishability attacks. *In a binary domain, fake data significantly increases the false positive rate, leading to a decrease in the estimated privacy loss ϵ_{emp}.* However, for a higher domain size, fake data has a lesser impact on the false positive rate, as the distinguishability attack has more rooms for errors. Overall, these nuanced relationships underscore the intricate interplay between domain size, the use of fake data and the tightness of local privacy loss estimation in the context of RS+FD protocols.

5.7 Case Study #5: Debugging a Python Implementation of UE Protocols

Finally, we show how our LDP-Auditor framework can also serve as a tool for verifying the correctness of LDP implementations. In our case study, we focused on the pure-LDP [46] package (version 1.1.2) and show that their UE protocols fail to meet the claimed level of ϵ-LDP. Our objective here is not to point out issues with respect to a particular code or library but rather to demonstrate the potentiality



(a) Domain size $k = 2$. **(b) Domain size $k = 100$.**

Figure 7: Theoretical ϵ (x-axis) versus estimated ϵ_{emp} (y-axis) using our LDP-Auditor framework comparing different number of attributes d for five RS+FD [5] protocols with domain sizes $k = 2$ and $k = 100$.

of our approach for verifying and debugging LDP protocols. Following a similar experimental setup as the one outlined in Section 5.2, Figure 8 presents a comparison of the theoretical ϵ values (x-axis) with the estimated ϵ_{emp} values (y-axis) using our LDP-Auditor framework. We consider different domain sizes k for both the SUE and OUE protocols, implemented in the pure-LDP package. The inconsistencies we found between the lower and upper bounds are highlighted within the orange rectangle.

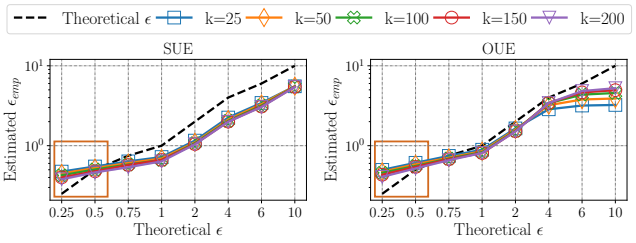


Figure 8: Theoretical ϵ (x-axis) versus estimated ϵ_{emp} (y-axis) using our LDP-Auditor framework comparing different domain sizes k for both SUE and OUE protocols, implemented in the pure-LDP package [46]. The orange rectangle highlights inconsistencies between the observed empirical privacy loss and the theoretical upper bound.

From Figure 8, it is clear that LDP-Auditor has detected inconsistencies between the lower and upper bounds, which are highlighted by the orange rectangle. After conducting an investigation into the pure-LDP code, we were able to identify the specific location of the implementation error. The error arises from the following steps in the `_perturb` function of the `UEClient` class:

(1) The user initializes a zero-vector $\mathbf{y} = [0, 0, \ldots, 0]$ of size k;
(2) The user samples indexes of values in $\mathbf{y}$ that will flip from 0 to 1 with probability q (as indicated in Equation (3)).
(3) With probability p (as indicated in Equation (3)), the index at position $\mathbf{y}_v$ (representing the user's true value) is flipped from 0 to 1.
(4) ***Missing step***: if $\mathbf{y}_v$ was set to 1 in step (2) but not in step (3), there should be a correction to revert it back to 0.

This was a simple mistake that ***was directly fixed by the authors [47] following our communication with them.*** However, it is crucial to emphasize that this minor error had implications for the ϵ-LDP guarantees. Specifically, the bit corresponding to the user's value was transmitted more time than intended, particularly in high privacy regimes. In mid to low privacy regimes, the bug might go unnoticed, given the already high probability of transmitting the bit as 1. This explains why LDP-Auditor failed to detect inconsistencies between the empirical and upper bounds for $\epsilon \geq 1$. In such cases, specialized tools designed for identifying DP violations, like DP-Sniper [17], would likely have been effective in detecting the bug. Therefore, we strongly encourage end-users of the pure-LDP package to update to the latest version 1.2.0.

6 CONCLUSION AND PERSPECTIVES

In this work, we have introduced the LDP-Auditor framework as a powerful tool for empirically estimating the privacy loss of LDP frequency estimation protocols. Our main LDP audit results provide new insights into the empirical local privacy loss in practical adversarial settings. Through several case studies, we have demonstrated the framework's effectiveness in identifying significant discrepancies between theoretical guarantees and empirical privacy loss. These findings contribute to a nuanced understanding of the challenges and considerations in the design and implementation of LDP mechanisms. As LDP continues to gain prominence in privacy-preserving data analysis, LDP-Auditor can serve as a valuable resource for practitioners and researchers aiming to assess and enhance the privacy guarantees of their systems.

Nevertheless, while we instantiated LDP-Auditor with distinguishability attacks on the user's value [9, 31], our future plans involve expanding the scope of LDP auditing to incorporate other adversarial analysis proposed in the literature, such as inference pool [33], data change detection [10] and re-identification attacks [9, 52]. We also aim and suggest extending LDP-Auditor to encompass a wider range of LDP applications (*e.g.*, mean estimation) as these may introduce unique challenges and considerations during the auditing process. Additionally, we aim at integrating the Neyman-Pearson lemma into LDP-Auditor's analysis to leverage its theoretical foundation to enhance the precision of our auditing framework. Lastly, one can envision utilizing LDP-Auditor as a means to establish a unified local privacy loss ϵ_{emp} when comparing mechanisms of different locally private definitions, such as d-privacy [20], α-PIE [52] and LDP [39].

ACKNOWLEDGMENTS

We thank Catuscia Palamidessi, Aurélien Bellet and Mathias Lécuyer for their helpful discussions and feedback throughout this project. The authors also deeply thank the anonymous PETS reviewers for their insightful suggestions. This work has been partially supported by the "ANR 22-PECY-0002" IPOP (Interdisciplinary Project on Privacy) project of the Cybersecurity PEPR. The work of Héber H. Arcolezi was partially supported by the European Research Council (ERC) project HYPATIA under the European Union's Horizon 2020 research and innovation programme. Grant agreement n. 835294. Sébastien Gambs is supported by the Canada Research Chair program as well as a Discovery Grant from NSERC.

REFERENCES

[1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. 2016. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 308–318. https://doi.org/10.1145/2976749.2978318

[2] Jayadev Acharya, Ziteng Sun, and Huanyu Zhang. 2019. Hadamard Response: Estimating Distributions Privately, Efficiently, and with Little Communication. In *Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (Proceedings of Machine Learning Research, Vol. 89)*, Kamalika Chaudhuri and Masashi Sugiyama (Eds.). PMLR, 1120–1129.

[3] Galen Andrew, Peter Kairouz, Sewoong Oh, Alina Oprea, H Brendan McMahan, and Vinith Suriyakumar. 2023. One-shot Empirical Privacy Estimation for Federated Learning. *arXiv preprint arXiv:2302.03098* (2023).

[4] Héber H. Arcolezi. 2024. https://github.com/hharcolezi/ldp-audit.

[5] Héber H. Arcolezi, Jean-François Couchot, Bechara Al Bouna, and Xiaokui Xiao. 2021. Random Sampling Plus Fake Data: Multidimensional Frequency Estimates With Local Differential Privacy. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. ACM, 47–57. https://doi.org/10.1145/3459637.3482467

[6] Héber H. Arcolezi, Jean-François Couchot, Bechara Al Bouna, and Xiaokui Xiao. 2021. Longitudinal Collection and Analysis of Mobile Phone Data with Local Differential Privacy. In *Privacy and Identity Management*, Michael Friedewald, Stefan Schiffner, and Stephan Krenn (Eds.). Springer International Publishing, Cham, 40–57. https://doi.org/10.1007/978-3-030-72465-8_3

[7] Héber H. Arcolezi, Jean-François Couchot, Bechara Al Bouna, and Xiaokui Xiao. 2022. Improving the utility of locally differentially private protocols for longitudinal and multidimensional frequency estimates. *Digital Communications and Networks* (July 2022). https://doi.org/10.1016/j.dcan.2022.07.003

[8] Héber H. Arcolezi, Jean-François Couchot, Sébastien Gambs, Catuscia Palamidessi, and Majid Zolfaghari. 2022. Multi-Freq-LDPy: Multiple Frequency Estimation Under Local Differential Privacy in Python. In *Computer Security – ESORICS 2022*, Vijayalakshmi Atluri, Roberto Di Pietro, Christian D. Jensen, and Weizhi Meng (Eds.). Springer Nature Switzerland, Cham, 770–775. https://doi.org/10.1007/978-3-031-17143-7_40

[9] Héber H. Arcolezi, Sébastien Gambs, Jean-François Couchot, and Catuscia Palamidessi. 2023. On the Risks of Collecting Multidimensional Data Under Local Differential Privacy. *Proc. VLDB Endow.* 16, 5 (jan 2023), 1126–1139. https://doi.org/10.14778/3579075.3579086

[10] Héber H. Arcolezi, Carlos A Pinzón, Catuscia Palamidessi, and Sébastien Gambs. 2023. Frequency Estimation of Evolving Data Under Local Differential Privacy. In *Proceedings of the 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28 - March 31, 2023*. OpenProceedings.org, 512–525. https://doi.org/10.48786/EDBT.2023.44

[11] Önder Askin, Tim Kutta, and Holger Dette. 2022. Statistical Quantification of Differential Privacy: A Local Approach. In *2022 IEEE Symposium on Security and Privacy (SP)*. 402–421. https://doi.org/10.1109/SP46214.2022.9833689

[12] Borja Balle. 2018. Analytic gaussian mechanism. https://github.com/BorjaBalle/analytic-gaussian-mechanism/blob/master/agm-example.py.

[13] Borja Balle, Gilles Barthe, and Marco Gaboardi. 2018. Privacy Amplification by Subsampling: Tight Analyses via Couplings and Divergences. In *Advances in Neural Information Processing Systems*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett (Eds.), Vol. 31. Curran Associates, Inc.

[14] Borja Balle and Yu-Xiang Wang. 2018. Improving the Gaussian Mechanism for Differential Privacy: Analytical Calibration and Optimal Denoising. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 80)*, Jennifer Dy and Andreas Krause (Eds.). PMLR, 394–403.

[15] Raef Bassily and Adam Smith. 2015. Local, Private, Efficient Protocols for Succinct Histograms. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing* (Portland, Oregon, USA) *(STOC '15)*. Association for Computing Machinery, New York, NY, USA, 127–135. https://doi.org/10.1145/2746539.2746632

[16] Karuna Bhaila, Wen Huang, Yongkai Wu, and Xintao Wu. 2024. Local Differential Privacy in Graph Neural Networks: a Reconstruction Approach. In *Proceedings of the 2024 SIAM International Conference on Data Mining (SDM)*. SIAM, 1–9. https://doi.org/10.1137/1.9781611978032.1

[17] Benjamin Bichsel, Samuel Steffen, Ilija Bogunovic, and Martin Vechev. 2021. DP-Sniper: Black-Box Discovery of Differential Privacy Violations using Classifiers. In *2021 IEEE Symposium on Security and Privacy (SP)*. 391–409. https://doi.org/10.1109/SP40001.2021.00081

[18] Mark Bun, Jelani Nelson, and Uri Stemmer. 2019. Heavy Hitters and the Structure of Local Privacy. *ACM Trans. Algorithms* 15, 4, Article 51 (oct 2019). https://doi.org/10.1145/3344722

[19] Tudor Cebere, Aurélien Bellet, and Nicolas Papernot. 2024. Tighter Privacy Auditing of DP-SGD in the Hidden State Threat Model. *arXiv preprint arXiv:2405.14457* (2024).

[20] Konstantinos Chatzikokolakis, Miguel E Andrés, Nicolás Emilio Bordenabe, and Catuscia Palamidessi. 2013. Broadening the scope of differential privacy using metrics. In *Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013. Proceedings 13*. Springer, 82–102.

[21] K. Chatzikokolakis, G. Cherubin, C. Palamidessi, and C. Troncoso. 2023. Bayes Security: A Not So Average Metric. In *2023 2023 IEEE 36th Computer Security Foundations Symposium (CSF) (CSF)*. IEEE Computer Society, Los Alamitos, CA, USA, 159–177. https://doi.org/10.1109/CSF57540.2023.00011

[22] Charles J Clopper and Egon S Pearson. 1934. The use of confidence or fiducial limits illustrated in the case of the binomial. *Biometrika* 26, 4 (1934), 404–413.

[23] Graham Cormode, Samuel Maddock, and Carsten Maple. 2021. Frequency estimation under local differential privacy. *Proceedings of the VLDB Endowment* 14, 11 (July 2021), 2046–2058. https://doi.org/10.14778/3476249.3476261

[24] José Serafim Costa Filho and Javam C Machado. 2023. FELIP: A local Differentially Private approach to frequency estimation on multidimensional datasets. In *Proceedings of the 26th International Conference on Extending Database Technology, EDBT 2023, Ioannina, Greece, March 28 - March 31, 2023*. OpenProceedings.org, 671–683. https://doi.org/10.48786/EDBT.2023.56

[25] Damien Desfontaines. 2021. A list of real-world uses of differential privacy. *Ted is writing things* (2021). https://desfontain.es/privacy/real-world-differential-privacy.html.

[26] Bolin Ding, Janardhan Kulkarni, and Sergey Yekhanin. 2017. Collecting Telemetry Data Privately. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 3571–3580.

[27] Zeyu Ding, Yuxin Wang, Guanhong Wang, Danfeng Zhang, and Daniel Kifer. 2018. Detecting Violations of Differential Privacy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. Association for Computing Machinery, New York, NY, USA, 475–489. https://doi.org/10.1145/3243734.3243818

[28] John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. 2013. Local Privacy and Statistical Minimax Rates. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. IEEE, 429–438. https://doi.org/10.1109/focs.2013.53

[29] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Theory of Cryptography*. Springer Berlin Heidelberg, 265–284. https://doi.org/10.1007/11681878_14

[30] Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science* 9, 3–4 (2014), 211–407.

[31] M. Emre Gursoy, Ling Liu, Ka-Ho Chow, Stacey Truex, and Wenqi Wei. 2022. An Adversarial Approach to Protocol Analysis and Selection in Local Differential Privacy. *IEEE Transactions on Information Forensics and Security* 17 (2022), 1785–1799. https://doi.org/10.1109/TIFS.2022.3170242

[32] Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. 2014. RAPPOR: Randomized Aggregatable Privacy-Preserving Ordinal Response. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security* (Scottsdale, Arizona, USA). ACM, New York, NY, USA, 1054–1067. https://doi.org/10.1145/2660267.2660348

[33] Andrea Gadotti, Florimond Houssiau, Meenatchi Sundaram Muthu Selva Annamalai, and Yves-Alexandre de Montjoye. 2022. Pool Inference Attacks on Local Differential Privacy: Quantifying the Privacy Guarantees of Apple's Count Mean Sketch in Practice. In *31st USENIX Security Symposium (USENIX Security 22)*. USENIX Association, Boston, MA, 501–518.

[34] Daniele Gorla, Louis Jalouzot, Federica Granese, Catuscia Palamidessi, and Pablo Piantanida. 2022. On the (Im) Possibility of Estimating Various Notions of Differential Privacy. *arXiv preprint arXiv:2208.14414* (2022).

[35] Mehmet Emre GÜRSOY. 2024. Longitudinal attacks against iterative data collection with local differential privacy. *Turkish Journal of Electrical Engineering*

and Computer Sciences 32, 1 (Feb. 2024), 198–218. https://doi.org/10.55730/1300-0632.4063

[36] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. 2020. Auditing Differentially Private Machine Learning: How Private is Private SGD?. In *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (Eds.), Vol. 33. Curran Associates, Inc., 22205–22216.

[37] Peter Kairouz, Keith Bonawitz, and Daniel Ramage. 2016. Discrete distribution estimation under local privacy. In *International Conference on Machine Learning*. PMLR, 2436–2444.

[38] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. The Composition Theorem for Differential Privacy. In *Proceedings of the 32nd International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 37)*, Francis Bach and David Blei (Eds.). PMLR, Lille, France, 1376–1385.

[39] Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. 2011. What Can We Learn Privately? *SIAM J. Comput.* 40, 3 (2011), 793–826. https://doi.org/10.1137/090756090

[40] Mishaal Kazmi, Hadrien Lautraite, Alireza Akbari, Mauricio Soroco, Qiaoyue Tang, Tao Wang, Sébastien Gambs, and Mathias Lécuyer. 2024. PANORAMIA: Privacy Auditing of Machine Learning Models without Retraining. *arXiv preprint arXiv:2402.09477* (2024).

[41] Hiroaki Kikuchi. 2022. Castell: Scalable Joint Probability Estimation of Multi-dimensional Data Randomized with Local Differential Privacy. *arXiv preprint arXiv:2212.01627* (2022).

[42] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. 2015. Numba: A LLVM-Based Python JIT Compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC* (Austin, Texas) *(LLVM '15)*. Association for Computing Machinery, New York, NY, USA, Article 7, 6 pages. https://doi.org/10.1145/2833157.2833162

[43] Ninghui Li, Wahbeh Qardaji, and Dong Su. 2012. On Sampling, Anonymization, and Differential Privacy or, k-Anonymization Meets Differential Privacy. In *Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security* (Seoul, Korea) *(ASIACCS '12)*. Association for Computing Machinery, New York, NY, USA, 32–33. https://doi.org/10.1145/2414456.2414474

[44] Fred Lu, Joseph Munoz, Maya Fuchs, Tyler LeBlond, Elliott Zaresky-Williams, Edward Raff, Francis Ferraro, and Brian Testa. 2022. A General Framework for Auditing Differentially Private Machine Learning. In *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 4165–4176.

[45] Yun Lu, Malik Magdon-Ismail, Yu Wei, and Vassilis Zikas. 2022. Eureka: A General Framework for Black-box Differential Privacy Estimators. Cryptology ePrint Archive, Paper 2022/1250. https://eprint.iacr.org/2022/1250.

[46] Samuel Maddock. 2021. pure-LDP. https://pypi.org/project/pure-ldp/.

[47] Samuel Maddock. 2023. Fix perturb bug in UEClient. https://github.com/Samuel-Maddock/pure-LDP/commit/fc622e338b565b9e6e7d75bc734e7859f6b6d2cc.

[48] Samuel Maddock, Alexandre Sablayrolles, and Pierre Stock. 2022. CANIFE: Crafting Canaries for Empirical Privacy Measurement in Federated Learning. *arXiv preprint arXiv:2210.02912* (2022).

[49] Pathum Chamikara Mahawaga Arachchige, Peter Bertok, Ibrahim Khalil, Dongxi Liu, Seyit Camtepe, and Mohammed Atiquzzaman. 2020. Local Differential Privacy for Deep Learning. *IEEE Internet of Things Journal* 7, 7 (2020), 5827–5842. https://doi.org/10.1109/JIOT.2019.2952146

[50] David McCandless, Tom Evans, Miriam Quick, Ella Hollowood, Christian Miles, Dan Hampson, and Duncan Geere. 2021. World's Biggest Data Breaches & Hacks. Available online: https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/ (accessed on 11 March 2023).

[51] Philipp Moritz, Robert Nishihara, Stephanie Wang, Alexey Tumanov, Richard Liaw, Eric Liang, Melih Elibol, Zongheng Yang, William Paul, Michael I. Jordan, and Ion Stoica. 2018. Ray: A Distributed Framework for Emerging AI Applications. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. USENIX Association, Carlsbad, CA, 561–577.

[52] Takao Murakami and Kenta Takahashi. 2021. Toward Evaluating Re-identification Risks in the Local Privacy Model. *Transactions on Data Privacy* 14, 3 (2021), 79–116.

[53] Milad Nasr, Jamie Hayes, Thomas Steinke, Borja Balle, Florian Tramèr, Matthew Jagielski, Nicholas Carlini, and Andreas Terzis. 2023. Tight Auditing of Differentially Private Machine Learning. *arXiv preprint arXiv:2302.07956* (2023).

[54] Milad Nasr, Shuang Songi, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlin. 2021. Adversary instantiation: Lower bounds for differentially private machine learning. In *2021 IEEE Symposium on security and privacy (SP)*. IEEE, 866–882.

[55] Thông T Nguyên, Xiaokui Xiao, Yin Yang, Siu Cheung Hui, Hyejin Shin, and Junbum Shin. 2016. Collecting and analyzing data from smart device users with local differential privacy. *arXiv preprint arXiv:1606.05053* (2016).

[56] Krishna Pillutla, Galen Andrew, Peter Kairouz, H Brendan McMahan, Alina Oprea, and Sewoong Oh. 2023. Unleashing the Power of Randomization in Auditing Differentially Private ML. *arXiv preprint arXiv:2305.18447* (2023).

[57] Xuebin Ren, Chia-mu Yu, Weiren Yu, Shusen Yang, Senior Member, Xinyu Yang, Julie A Mccann, Philip S Yu, and Life Fellow. 2018. LoPub : High-Dimensional Crowdsourced Data. 13, 9 (2018), 2151–2166. https://doi.org/10.1109/TIFS.2018.2812146

[58] Thomas Steinke, Milad Nasr, and Matthew Jagielski. 2023. Privacy Auditing with One (1) Training Run. *arXiv preprint arXiv:2305.08846* (2023).

[59] Apple Differential Privacy Team. 2017. Learning with privacy at scale. https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf, (accessed January 2023).

[60] Florian Tramer, Andreas Terzis, Thomas Steinke, Shuang Song, Matthew Jagielski, and Nicholas Carlini. 2022. Debugging differential privacy: A case study for privacy auditing. *arXiv preprint arXiv:2202.12219* (2022).

[61] Florian Turati, Karel Kubicek, Carlos Cotrini, and David Basin. 2023. Locality-Sensitive Hashing Does Not Guarantee Privacy! Attacks on Google's FLoC and the MinHash Hierarchy System. *Proceedings on Privacy Enhancing Technologies* 2023, 4 (Oct. 2023), 117–131. https://doi.org/10.56553/popets-2023-0101

[62] Stefan van der Walt, S. Chris Colbert, and Gael Varoquaux. 2011. The NumPy Array: A Structure for Efficient Numerical Computation. *Computing in Science & Engineering* 13, 2 (2011), 22–30. https://doi.org/10.1109/MCSE.2011.37

[63] Israel De Castro Vidal, André Luís da Costa Mendonça, Franck Rousseau, and Javam De Castro Machado. 2020. ProTECting: An Application of Local Differential Privacy for IoT at the Edge in Smart Home Scenarios. In *Anais XXXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2020)*. Sociedade Brasileira de Computação. https://doi.org/10.5753/sbrc.2020.12308

[64] Ning Wang, Xiaokui Xiao, Yin Yang, Jun Zhao, Siu Cheung Hui, Hyejin Shin, Junbum Shin, and Ge Yu. 2019. Collecting and Analyzing Multidimensional Data with Local Differential Privacy. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE. https://doi.org/10.1109/icde.2019.00063

[65] Shaowei Wang, Liusheng Huang, Pengzhan Wang, Yiwen Nie, Hongli Xu, Wei Yang, Xiang-Yang Li, and Chunming Qiao. 2016. Mutual information optimally local private discrete distribution estimation. *arXiv preprint arXiv:1607.08025* (2016).

[66] Tianhao Wang, Jeremiah Blocki, Ninghui Li, and Somesh Jha. 2017. Locally Differentially Private Protocols for Frequency Estimation. In *26th USENIX Security Symposium (USENIX Security 17)*. USENIX Association, Vancouver, BC, 729–745.

[67] Tianhao Wang, Ninghui Li, and Somesh Jha. 2018. Locally Differentially Private Frequent Itemset Mining. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE. https://doi.org/10.1109/sp.2018.00035

[68] Tianhao Wang, Ninghui Li, and Somesh Jha. 2021. Locally Differentially Private Heavy Hitter Identification. *IEEE Transactions on Dependable and Secure Computing* 18, 2 (March 2021), 982–993. https://doi.org/10.1109/tdsc.2019.2927695

[69] Teng Wang, Jun Zhao, Zhi Hu, Xinyu Yang, Xuebin Ren, and Kwok-Yan Lam. 2021. Local Differential Privacy for data collection and analysis. *Neurocomputing* 426 (Feb. 2021), 114–133. https://doi.org/10.1016/j.neucom.2020.09.073

[70] Stanley L. Warner. 1965. Randomized Response: A Survey Technique for Eliminating Evasive Answer Bias. *J. Amer. Statist. Assoc.* 60, 309 (March 1965), 63–69. https://doi.org/10.1080/01621459.1965.10480775

[71] Julia Carrie Wong. 2019. Facebook to be fined $5bn for Cambridge Analytica privacy violations – reports. Available online: https://www.theguardian.com/technology/2019/jul/12/facebook-fine-ftc-privacy-violations (accessed on 11 March 2023).

[72] Haonan Wu, Ruisheng Ran, Shunshun Peng, Mengmeng Yang, and Taolin Guo. 2023. Mining frequent items from high-dimensional set-valued data under local differential privacy protection. *Expert Systems with Applications* 234 (Dec. 2023), 121105. https://doi.org/10.1016/j.eswa.2023.121105

[73] Min Ye and Alexander Barg. 2018. Optimal Schemes for Discrete Distribution Estimation Under Locally Differential Privacy. *IEEE Transactions on Information Theory* 64, 8 (2018), 5662–5676. https://doi.org/10.1109/TIT.2018.2809790

[74] Emre Yilmaz, Mohammad Al-Rubaie, and J. Morris Chang. 2020. Naive Bayes Classification under Local Differential Privacy. In *2020 IEEE 7th International Conference on Data Science and Advanced Analytics (DSAA)*. IEEE. https://doi.org/10.1109/dsaa49011.2020.00081

[75] Zhikun Zhang, Tianhao Wang, Ninghui Li, Shibo He, and Jiming Chen. 2018. CALM: Consistent adaptive local marginal for marginal release under local differential privacy. *Proceedings of the ACM Conference on Computer and Communications Security* (2018), 212–229. https://doi.org/10.1145/3243734.3243742

A SUMMARY OF NOTATIONS

The main notation used in this paper is summarized in Table 1.

Symbol	Description
$[a]$	Set of integers $\{1, 2, 3, \ldots, a\}$.
$\mathbf{a}_i$	i-th coordinate of vector $\mathbf{a}$.
V	Data domain.
k	Domain size $k = \lvert V \rvert$.
n	Number of users.
ϵ	Theoretical privacy loss.
δ	Maximum probability that privacy loss exceeds ϵ.
ϵ_{emp}	Empirical privacy loss.
ϵ_{OPT}	Upper bound on Monte Carlo privacy loss.
$\mathcal{M}$	(ϵ, δ)-LDP mechanism.
$\mathcal{A}$	Distinguishability attack.
$\mathcal{A}^L$	Distinguishability attack in longitudinal study.
$\mathcal{A}^{\text{RS+FD}}$	Distinguishability attack on RS+FD protocols.
$\mathcal{A}_{\mathcal{M}}$	Distinguishability attack of mechanism $\mathcal{M}$.
$\mathbb{1}_{\mathcal{M}}$	Support set of mechanism $\mathcal{M}$.
T	Number of trials.
α	Confidence level.
d	Number of attributes $d \geq 2$.
τ	Number of data collections.

Table 1: Symbols and Notations.

B APPROXIMATE (ϵ, δ)-LDP PROTOCOLS

Approximate GRR (AGRR) [69]. Similar to GRR in Section 3.1, given a value $v \in V$, AGRR(v) outputs the true value v with probability p, and any other value $v' \in V \setminus \{v\}$, otherwise. More formally:

$$\Pr[\text{AGRR}(v) = y] = \begin{cases} p = \frac{e^\epsilon + (k-1)\delta}{e^\epsilon + k - 1} & \text{if } y = v, \\ q = \frac{1-\delta}{e^\epsilon + k - 1} & \text{if } y \neq v, \end{cases} \tag{9}$$

in which $y \in V$ is the perturbed value sent to the server. From Equation (9), $\Pr[y = v] > \Pr[y = v']$ for all $v' \in V \setminus \{v\}$. Thus, the attack strategy $\mathcal{A}_{\text{AGRR}}$ is equivalent to $\mathcal{A}_{\text{GRR}}$, *i.e.*, to predict $\hat{v} = y$.

Approximate SUE (ASUE) [69]. Similar to the SUE protocol [32] in Section 3.1, ASUE encode the user's input data $v \in V$, as a one-hot k-dimensional vector. The obfuscation function of ASUE randomizes the bits from $\mathbf{v}$ independently to generate $\mathbf{y}$ as follows:

$$\forall i \in [k]: \quad \Pr[\mathbf{y}_i = 1] = \begin{cases} p = \frac{e^\epsilon - \sqrt{e^\epsilon (1-\delta) + \delta}}{e^\epsilon - 1}, & \text{if } \mathbf{v}_i = 1, \\ q = \frac{\sqrt{e^\epsilon (1-\delta) + \delta} - 1}{e^\epsilon - 1}, & \text{if } \mathbf{v}_i = 0, \end{cases} \tag{10}$$

in which $\mathbf{y}$ is sent to the server. As for UE protocols, with $\mathbf{y}$, the adversary can construct the subset of all values $v \in V$ that are set to 1, *i.e.*, $\mathbb{1}_{\text{AUE}} = \{v | \mathbf{y}_v = 1\}$. Then, the attack strategy $\mathcal{A}_{\text{ASUE}}$ is equivalent to $\mathcal{A}_{\text{AUE}}$:

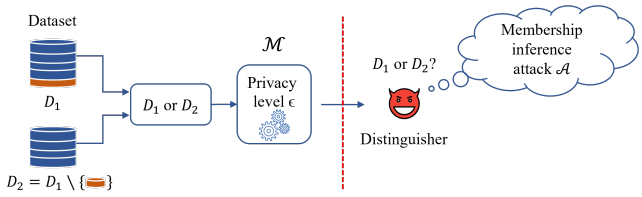
- $\mathcal{A}^0_{\text{ASUE}}$ is a random choice $\hat{v} = \text{Uniform}([k])$, if $\mathbb{1}_{\text{AUE}} = \emptyset$;
- $\mathcal{A}^1_{\text{ASUE}}$ is a random choice $\hat{v} = \text{Uniform}(\mathbb{1}_{\text{AUE}})$, otherwise.

Approximate LH (ALH) [69]. Similar to the LH protocols [15, 66] in Section 3.1, ALH uses a hash function $\text{H} \in \mathscr{H}$ to map the input data $v \in V$ to a new domain of size $g \geq 2$, and then apply AGRR to the hashed value $h = \text{H}(v)$. In particular, the ALH reporting mechanism is $\text{ALH}(v) := \langle \text{H}, \text{AGRR}(h) \rangle$, in which AGRR is given in Equation (9) while operating on the new domain $[g]$. The two variants of ALH protocols are: (1) Approximate BLH (ABLH), which sets $g = 2$ and (2) Approximate OLH (AOLH), which sets $g = \frac{-3e^\epsilon\delta - \sqrt{e^\epsilon - 1}\sqrt{(1-\delta)(e^\epsilon + \delta - 9e^\epsilon\delta - 1)} + e^\epsilon + 3\delta - 1}{2\delta}$. Each user reports the hash function and obfuscated value $\langle \text{H}, y \rangle$ to the server. With these elements, the adversary can construct the subset of all values $v \in V$ that hash to y, *i.e.*, $\mathbb{1}_{\text{ALH}} = \{v | \text{H}(v) = y\}$. Then, the attack strategy $\mathcal{A}_{\text{ALH}}$ is equivalent to $\mathcal{A}_{\text{LH}}$:

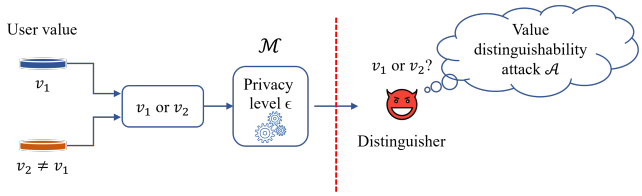
- $\mathcal{A}^0_{\text{ALH}}$ is a random choice $\hat{v} = \text{Uniform}([k])$, if $\mathbb{1}_{\text{ALH}} = \emptyset$;
- $\mathcal{A}^1_{\text{ALH}}$ is a random choice $\hat{v} = \text{Uniform}(\mathbb{1}_{\text{ALH}})$, otherwise.

C ADVERSARIAL PRIVACY GAME

Figure 9 provides a comparative illustration of the adversarial privacy game in central and local differential privacy frameworks, highlighting scenarios of membership inference and value distinguishability attacks, respectively.



(a) Central DP with membership inference attack.



(b) Local DP with value distinguishability attack.

Figure 9: Comparison of the adversarial privacy game between the central and local DP settings.

D CLOPPER-PEARSON INTERVAL

The Clopper-Pearson method [22] is a statistical technique used to calculate exact confidence intervals for the success probability in binomial distributions. This method is known for its conservative nature, ensuring that the confidence interval computed does not rely on any asymptotic approximations and is therefore valid regardless of the sample size. Given x successes in T trials, the Clopper-Pearson interval computes the lower and upper confidence limits for the true probability of success, based on the beta distribution's cumulative density function. Specifically, the Clopper-Pearson confidence interval is computed as follows:

$$\left[\mathfrak{B}\left(\frac{\alpha}{2}; x, T - x + 1\right), \mathfrak{B}\left(1 - \frac{\alpha}{2}; x + 1, T - x\right) \right], \tag{11}$$

in which $\mathfrak{B}$ denotes the beta distribution quantile function, x is the number of observed successes, T is the total number of trials, α

represents the significance level and $\mathfrak{B}(p; z, w)$ is the p-th quantile from a beta distribution with shape parameters z and w. This exact method is crucial in our LDP auditing framework, as it allows us to establish the lower and upper bounds for the true positive rate and false positive rate of Equation (8) with high confidence, ensuring that our empirical privacy loss estimations are both accurate and robust.

E PROOF OF THEOREM 1

Proof of Theorem 1. First, the guarantee of the Clopper-Pearson confidence intervals is that, with probability at least $1 - \alpha$, $\hat{p}_0 \leq p_0$ and $\hat{p}_1 \geq p_1$, which implies $p_0/p_1 \geq \hat{p}_0/\hat{p}_1$. Second, if $\mathcal{M}$ is (ϵ, δ)-LDP, then we would have $p_0 \leq p_1 e^{\epsilon} + \delta$, meaning $\mathcal{M}$ is not (ϵ', δ)-LDP for any $\epsilon' < \ln((p_0 - \delta)/p_1)$. Combining the two statements, $\mathcal{M}$ is not ϵ' for any $\epsilon' < \ln((\hat{p}_0 - \delta)/\hat{p}_1) = \epsilon_{emp}$. □

F MEMOIZATION-BASED LDP PROTOCOLS

As mentioned in Section 4.2, in longitudinal studies, the privacy loss is linear on the number of data collections τ following the DP sequential composition. This accumulation allows attackers to employ "averaging attacks" to more easily distinguish a user's true value among the noisy data. To counteract this, renowned LDP mechanisms for longitudinal studies, such as RAPPOR [32] and dBitFlipPM [26], incorporate a *memoization-based* strategy.

One way to employ memoization is to memorize an obfuscated value $y = \mathcal{M}(v)$ and consistently reuse it throughout time [6, 26]. Specifically, at each time $t \in [\tau]$, the user reports the memorized y, which satisfies ϵ-LDP. Note that as there is only a single obfuscation round, our LDP-Auditor operates equivalently to auditing in a single data collection scenario (*i.e.*, Algorithm 1).

An alternative memoization technique involves re-using the memorized obfuscated value $y = \mathcal{M}(v)$ as the input for a subsequent round of obfuscation [7, 9, 32, 63]. This means that at each time $t \in [\tau]$ the user reports $y^t = \mathcal{M}(y)$; note that the input to $\mathcal{M}$ is an already obfuscated value y. In this setting, there are two levels of privacy guarantees [32]: ϵ_1, which is the privacy level of the first report $y^1 = \mathcal{M}(y)$ following the second obfuscation round, and ϵ_∞, which is the privacy guarantee offered by the first obfuscation round that generated y. More precisely, $y = \mathcal{M}(v)$ satisfy ϵ_∞-LDP because it establishes the upper bound for the privacy leakage as an adversary could only recover y instead of v after executing an "averaging attack" across an indefinite number of reports $y^1, y^2, \ldots, y^\infty$. Consequently, our LDP-Auditor framework described in Algorithm 1 can be deployed directly to estimate an empirical privacy loss ϵ_{emp} against the theoretical upper bound ϵ_1 for a single data collection. For $t \to \infty$ data collections, the theoretical upper bound becomes ϵ_∞, for which the distinguishability attack in longitudinal study $\mathcal{A}^L$ outlined in Algorithm 2 should be applied. In other words, while in Section 5.5 the upper bound is $\tau\epsilon$-LDP, for memoization-based mechanisms with two obfuscation rounds, the upper bound is ϵ_∞-LDP.

G ADDITIONAL EXPERIMENTS

G.1 Case Study #1: Auditing the Impact of δ

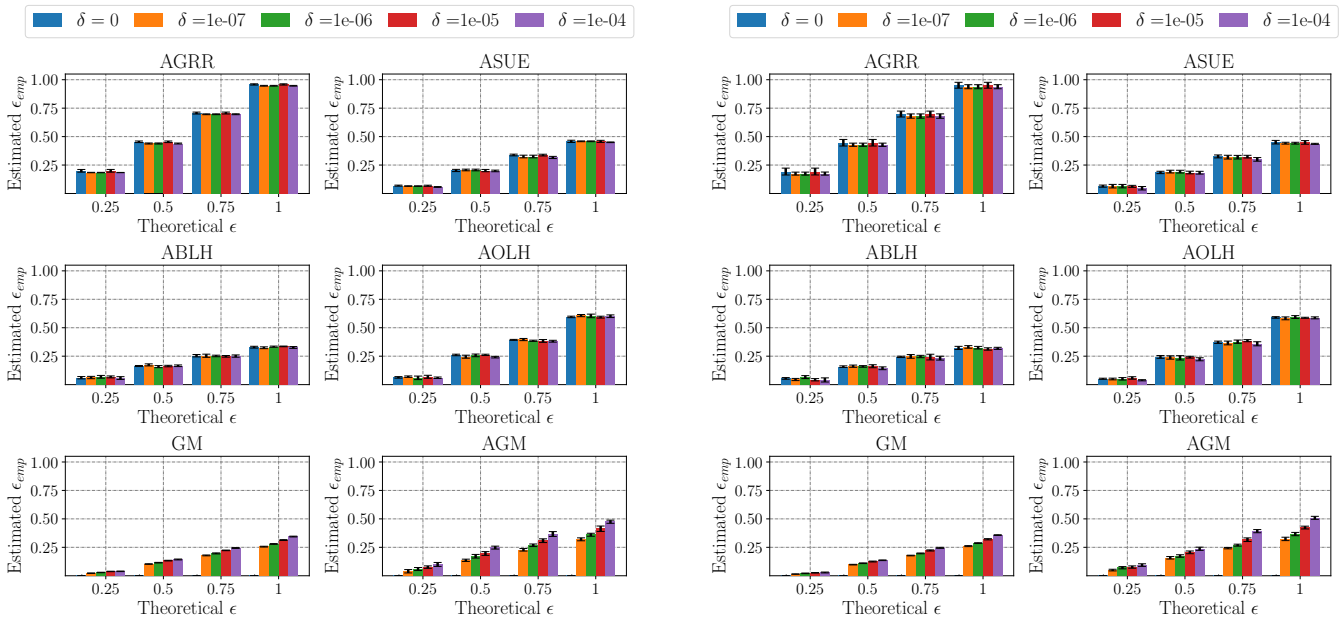
Following the experimental setup detailed in Section 5.3, Figure 10 illustrates the theoretical ϵ values (x-axis) versus estimated ϵ_{emp} values (y-axis) when varying the δ parameter and domain size $k \in \{100, 150\}$, using our LDP-Auditor framework. Note that for both GM and AGM protocols, there is no ϵ_{emp} value when $\delta = 0$, as these protocols do not have pure ϵ-LDP variations. Finally, a similar trend as in Figure 4 can be observed in Figure 10, for which the discussion in Section 5.3 is equally applicable to these results.

G.2 Case Study #3: Auditing the LDP Sequential Composition in Longitudinal Studies

Following the experimental setup detailed in Section 5.5, Figure 11 illustrates the estimated ϵ_{emp} values (y-axis) for the eight ϵ-LDP and both GM and AGM (ϵ, δ)-LDP protocols according to the the number of data collections τ (x-axis), per report ϵ and domain size $k \in \{25, 50\}$, using our LDP-Auditor framework. Notice that a similar trend as in Figure 6 can be observed in Figure 11, for which the discussion in Section 5.5 is equally applicable to these results.

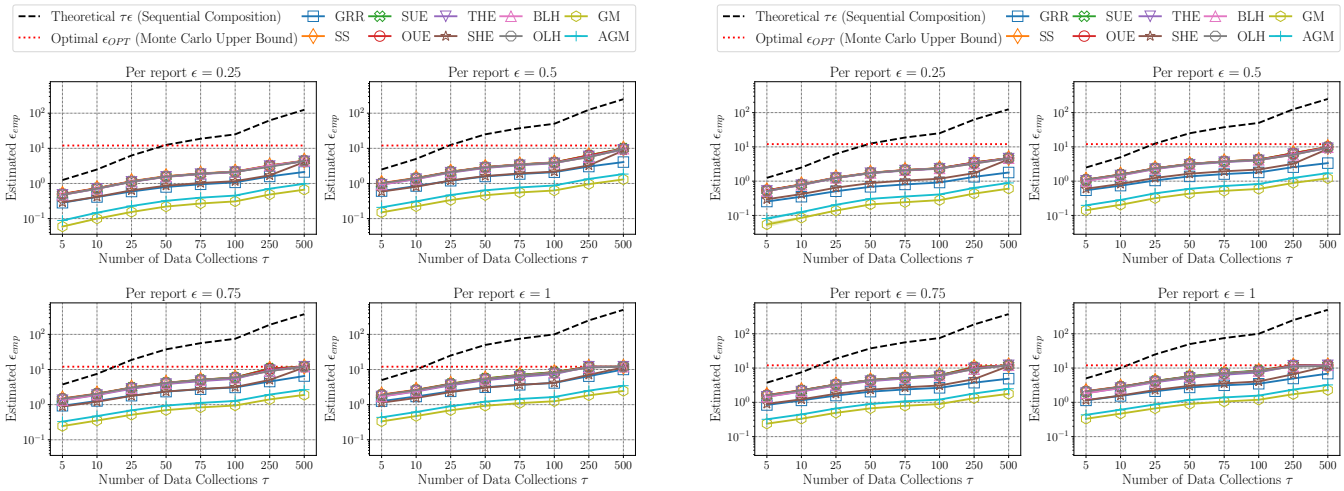
G.3 Case Study #4: LDP Auditing with Multidimensional Data

Following the experimental setup detailed in Section 5.6, Figure 12 illustrates the comparison of theoretical ϵ values (x-axis) with estimated ϵ_{emp} values (y-axis) for the five RS+FD protocols, based on the number of attributes d and domain size $k \in \{25, 50\}$, utilizing our LDP-Auditor framework. Notice that a similar trend as in Figure 7 can be observed in Figure 12, for which the discussion in Section 5.6 is equally applicable to these results.



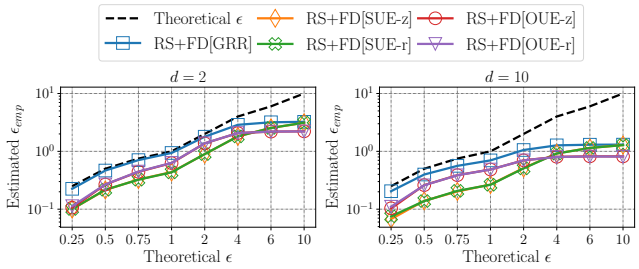
(a) Domain size $k = 100$. **(b) Domain size $k = 150$.**

Figure 10: Theoretical ϵ values (x-axis) versus estimated ϵ_{emp} values (y-axis) using our LDP-Auditor framework. We assess different privacy guarantees for six (ϵ, δ)-LDP protocols across domain sizes $k \in \{100, 150\}$. The special case $\delta = 0$ corresponds to pure ϵ-LDP, for which GM and AGM do not satisfy.

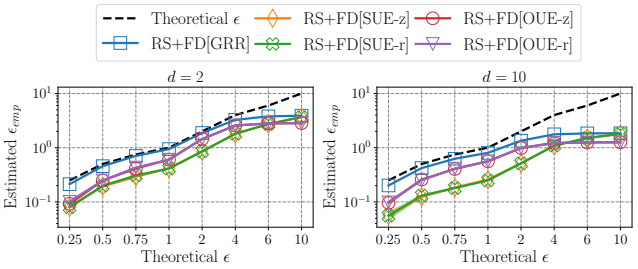


(a) Domain size $k = 25$. **(b) Domain size $k = 50$.**

Figure 11: Estimated ϵ_{emp} (y-axis) versus the number of data collections τ (x-axis) using our LDP-Auditor framework for different domain sizes $k \in \{25, 50\}$. We vary the per report ϵ-LDP guarantee for the following LDP frequency estimation protocols: GRR, SS, SUE, OUE, BLH, OLH, SHE, THE, GM and AGM. For both approximate-LDP protocols, namely GM and AGM, $\delta = 1e^{-5}$.



(a) Domain size $k = 25$.



(b) Domain size $k = 50$.

Figure 12: Theoretical ϵ (x-axis) versus estimated ϵ_{emp} (y-axis) using our LDP-Auditor framework comparing different number of attributes d for five RS+FD [5] protocols with domain sizes $k = 25$ and $k = 50$.