

A Unified View of Frequency Estimation and their Attacks on Local Differential Privacy

Al Mehdi Saadat Chowdhury* Dhaval Pankaj Tanna† Deepak Vellanki‡
Chirag Manjeshwar§

Abstract

Needs: What Problem, Summary of Results.

Expectation: understand the problem, the overview of the survey construction/protocol (pros and cons), the security assumption (e.g. Decisional Diffie–Hellman assumption), and the protocol’s performance.

1 Introduction

Generating meaningful statistical summaries about a population without revealing information about any individual is the central goal of privacy-preserving data analysis. Since its introduction, Differential Privacy (DP) has become the gold-standard framework for analyzing sensitive data while providing rigorous privacy guarantees. For any randomized algorithm M , DP is defined [3] as the following:

Definition 1.1 (Differential Privacy). Consider any database x as a collection of records taken from a universe \mathcal{X} , and is represented by their histograms: $x \in \mathbb{N}^{|\mathcal{X}|}$ in which each entry x_i represents the number of elements in x of type $i \in \mathcal{X}$. A randomized algorithm \mathcal{M} with domain $\mathbb{N}^{|\mathcal{X}|}$ is (ϵ, δ) -differentially private if for all $\mathcal{S} \subseteq Range(\mathcal{M})$ and for all $x, y \in \mathbb{N}^{|\mathcal{X}|}$ such that $\|x - y\|_1 \leq 1$:

$$Pr[\mathcal{M}(x) \in \mathcal{S}] \leq e^\epsilon Pr[\mathcal{M}(y) \in \mathcal{S}] + \delta$$

Stronger privacy guarantee is achieved by using smaller privacy loss bound parameter ϵ ; the parameter δ represents the probability that the guarantee fails to hold.

DP requires a central curator who collects the dataset and perturbs it to preserve privacy. This not only creates legal, ethical, and technical burden on the curator, but also the privacy itself becomes vulnerable if the curator is compromised. Local differential privacy (LDP) can solve this issue by asking each user to encrypt their data before sending to the curator. The only job of the curator remains is to aggregate the data from all users.

Definition 1.2 (Local Differential Privacy [7]). An algorithm \mathcal{M} satisfies ϵ -local differential privacay (ϵ -LDP), where $\epsilon \geq 0$, iff for any input v_1 and v_2 , we have:

$$\forall y \in Range(\mathcal{M}) : Pr[\mathcal{M}(v_1) = y] \leq e^\epsilon Pr[\mathcal{M}(v_2) = y]$$

LDP can be ensured by following three protocols. The **Encode** protocol takes an input value v and outputs an encoded value x . The **Perturb** protocol returns a noisy version of the encoded x as $y = Perturb(Encode(v))$. The **Aggregate** protocol takes perturbed values from all users and returns any required aggregate information. The first two protocols, **Encode** and **Perturb**, are executed by the user (we will combine them into one as **PE(v)**), and the curator executes **Aggregate**.

One of the most basic goals of the **Aggregate** task is *frequency estimation*. Given the input domain $[d] = \{1, 2, \dots, d\}$ of all users, frequency estimation seeks to estimate how many users have a given value $i \in [d]$. Since it serves as a building block for most **Aggregate** tasks, improving frequency estimation can significantly benefit other protocols, making it a critical problem.

Many popular methods for frequency estimation exists, including Google’s RAPPOR [4], Samsung’s Harmony [5], Direct Encoding [6], k-Randomized Response [2], Optimized Unary Encoding [7], Optimized Local Hashing [7]. In this work, we organize and review some of these estimation methods, identify which of these are relatively better, and show how attacks can be developed [1] against these better protocols.

*CSE PhD 3rd year

†CSE Master’s 2nd year

‡CSE Master’s 2nd year

§CSE Master’s 2nd year

2 Problem Setup and Threat Model

2.1 Problem Setup:

Because many protocols for frequency estimation had already appeared in the literature—each differing in how they perform the steps of **Encode**, **Perturb**, and **Aggregate**—a unified framework was needed to describe them consistently. Such a unified view, known as pure LDP, was introduced in [7] and is defined below.

Definition 2.1 (Pure LDP). Consider a function *Support* which maps each possible y to a set of input values that y supports. A protocol given by *PE* and *Support* is pure iff for all v_1

$$\begin{aligned} \Pr[\text{PE}(v_1) \in \{y | v_1 \in \text{Support}(y)\}] &= p^*, \\ \forall_{v_2 \neq v_1} \Pr[\text{PE}(v_2) \in \{y | v_1 \in \text{Support}(y)\}] &= q^* \end{aligned}$$

such that $p^* > q^*$. Intuitively, the probability that the perturbed encoded value of v_1 mapped to its own support set should be more than it is mapped to a different input’s support set.

A nice organization of protocols can be obtained by casting these to the framework of pure LDP protocols, based on how each protocol encodes an input:

- **Direct Encoding:** When no encoding is applied.
- **Histogram Encoding:** An input v is encoded as histogram for $[d]$ possible values. Adding noise from Laplace Distribution becomes the perturbation phase.
- **Unary Encoding:** An input v is encoded as a length- d vector. Perturbation is done by parameters p^* and q^* .
- **Local Hashing:** An input v is encoded by choosing at random a hash function H from a universe of hash function family \mathcal{H} , and then computing $(H, H(v))$. Perturbation is done by parameters p^* and q^* .

In this work, we consider the **kRR**, **OUE**, and **OLH** protocols—representing Direct Encoding, Unary Encoding, and Local Hashing, respectively—for comparison, and we propose attack strategies for each. Below, we provide a precise formulation of the attack problem we consider.

Problem Formulation 2.1: Attack on Frequency Estimation under Pure LDP

In this work, we consider **targeted** attacks (as opposed to untargeted attacks), and the goal of the attack is to increase the estimated frequency of the attacker-chosen target items. Specifically, we assume the system has n genuine users, and the attacker can inject m fake users (total users = $n + m$). The attacker considers a set $T = \{t_1, t_2, \dots, t_r\}$ of r target items. The goal is to increase the frequency of each t_i .

2.2 Threat Model:

Attacker’s Assumption: We assume the attacker can inject m fake users to the system. We also assume the attacker has access to the **Encode** and **Perturb** protocols, because these are executed locally on the user’s side. As a result, the attacker knows about the domain size d , the encoded space \mathcal{D} , and the support set $\{y | v \in \text{Support}(y)\}$ for each perturbed value $y \in \mathcal{D}$. We use \mathbf{Y} to denote the set of crafted perturbed values for the fake users.

Attacker’s Goal: For a set of attacker specified items $T = \{t_1, t_2, \dots, t_r\}$, the goal is to increase the estimated frequency of each t_i . Suppose $\hat{f}_{t,b}$ and $\hat{f}_{t,a}$ are the frequencies estimated for target item t before and after an attack. Then $\Delta \hat{f}_t = \hat{f}_{t,a} - \hat{f}_{t,b}, \forall t \in T$ is defined as the frequency gain for a target item t . The overall gain G is defined as: $G(\mathbf{Y}) = \sum_{t \in T} \mathbb{E}[\Delta \hat{f}_t]$, and the attacker’s goal is to maximize this overall gain:

$$\max_{\mathbf{Y}} G(\mathbf{Y}) \tag{1}$$

Attack Types: We use three attacks:

- **Random Perturbed-value Attack (RPA):** Select a perturbed value from \mathcal{D} uniformly at random for each fake user (without considering any target item t) and send it to the aggregator.
- **Random Item Attack (RIA):** Select a target item t from T uniformly at random for each fake user, and encode, perturb, and send it to the aggregator.
- **Maximal Gain Attack (MGA):** Solve the optimization problem of 1 to craft the perturbed values, and send it to the aggregator.

3 Theory/Construction/Analysis

3.1 Frequency Estimation Techniques:

To facilitate a rigorous comparison, we detail the construction of the three representative frequency estimation protocols under the framework of Pure LDP as unified by Wang et al. [7].

- **k-Randomized Response (kRR):** kRR is the direct generalization of the classic Randomized Response technique to a domain of size d . As described in [7], for an input value v , the user reports the true value v with probability $p = \frac{e^\epsilon}{e^\epsilon + d - 1}$ and reports a different value $v' \neq v$ chosen uniformly at random with probability $1 - p$. While simple to implement, the probability of reporting the truth decreases as the domain size d increases, necessitating a larger correction factor during aggregation to obtain an unbiased estimate.
- **Optimized Unary Encoding (OUE):** To address the dependency on d found in kRR, OUE encodes the input v into a binary vector of length d , where only the v -th bit is 1 and all other bits are 0 [7]. Each bit is then perturbed independently. Unlike Symmetric Unary Encoding (SUE), OUE optimizes the perturbation parameters to minimize variance for expected low-frequency inputs. Specifically, it sets the probability of preserving a 1 as $p = 0.5$ and the probability of flipping a 0 to a 1 as $q = \frac{1}{e^\epsilon + 1}$. This allows the variance to remain constant regardless of the domain size d .
- **Optimized Local Hashing (OLH):** While OUE provides variance independent of d , it incurs a high communication cost of $\Theta(d)$ [7]. OLH resolves this by mapping the input v to a smaller domain size g using a random hash function H chosen from a universal family [7]. The user computes $y = H(v)$ and perturbs y using the standard mechanism on the reduced domain g . The user reports the pair $\langle H, y \rangle$. Wang et al. [7] derive that the optimal domain size is $g \approx e^\epsilon + 1$, which mathematically aligns the variance of OLH with OUE while reducing communication cost from $\Theta(d)$ to $\Theta(n)$.

3.2 Attacks on Frequency Estimation:

We analyze the vulnerability of these protocols against the Maximal Gain Attack (MGA) proposed by Cao et al. [1]. In this model, the attacker controls m fake users and aims to maximize the estimated frequency of a target item set T .

- **Attacking kRR:** The attack strategy against kRR is straightforward but highly damaging in large domains. As detailed in [1], fake users simply report the target item $t \in T$ as their output. The aggregator, assuming the data follows the kRR noise distribution, applies an inverse transformation that scales the count by a factor proportional to d . Consequently, for large domain sizes, even a small number of fake reports results in a massive amplification of the estimated frequency for t , as the system overcompensates for the assumed high noise level.
- **Attacking OUE:** For Unary Encoding protocols, the attacker exploits the independent perturbation of bits. To execute the MGA against OUE, fake users construct a "poisoned" bit vector [1]. To promote a target t , the fake users deterministically set the t -th bit to 1 (the supported value). Depending on the specific gain formulation, the attacker may also manipulate the non-target bits to further statistically distinguish the fake inputs from genuine noise, although the primary gain is driven by the "support" of the target index.

- **Attacking OLH:** Since OLH relies on hashing, the attack leverages hash collisions. As described by Cao et al. [1], for a target t , the fake user explores the family of hash functions to find a specific function H and a perturbed value y such that $y = H(t)$. By reporting the pair $\langle H, y \rangle$, the fake user guarantees that the aggregator's decoding step will increment the count for t (along with other colliding values). This allows the attacker to inject bias into the estimation of t indistinguishable from valid hash reports.

4 Evaluation

4.1 Evaluation of Frequency estimation Techniques

Based on the problem setup and definitions in section 2, we will now evaluate the theoretical limits. To do this we utilize the pure LDP variance formula derived by Wang et al.[7]. This formula servers as our ruler, allowing us to rigorously compare the accuracy of each protocol. For any pure LDP protocol, the approximate variance of the frequency estimator is given by:

$$\text{Var}^* [\tilde{c}(i)] = \frac{nq^*(1 - q^*)}{(p * -q^*)^2} \quad (2)$$

We apply this ruler to the three protocols to demonstrate why optimization is necessary.

4.1.1 Analysis of k-Randomized Response(kRR)

For kRR, the probability of reporting the true value is $p = \frac{e^\epsilon}{e^\epsilon + d - 1}$. Substituting the parameters for kRR in Eq. 2 yields:

$$\text{Var}^* [\tilde{c}_{\text{DE}}(i)] \approx n \cdot \frac{d - 2 + e^\epsilon}{(e^\epsilon - 1)^2} \quad (3)$$

The variance scales linearly with the domain size d . For a domain size of $d = 10^5$, the noise overwhelms the signal. Hence kRR is theoretically optimal only for a small domain.

4.1.2 Analysis of Optimized Unary Encoding(OUE)

Wang et al. [7] that by encoding data into vectors and optimizing the perturbation parameters independently ($p = 0.5$, $q = \frac{1}{e^\epsilon + 1}$) the variance can be decoupled from the domain size. Subtracting these optimized parameters into Eq.2 yields:

$$\text{Var}^* [\tilde{c}_{\text{OUE}}(i)] = n \cdot \frac{4e^\epsilon}{(e^\epsilon - 1)^2} \quad (4)$$

Unlike kRR, this variance is independent of d making it highly accurate for large domains. However, it comes with a tradeoff. While accurate, OUE requires the user to transmit a bit vector of size d . For $d = 10^6$, this communication overhead is prohibitive($\Theta(d)$).

4.1.3 Analysis of Optimized Local Hashing(OLH)

OLH represents the solution to the communication/utility conflict. Wang et al. [7] treated the hash domain size g as a variable in the variance equation and minimized it by taking a partial derivative with respect to g , deriving the optimal size $g \approx e^\epsilon + 1$. Plugging this optimal value of g back in Eq.2 yields:

$$\text{Var}^* [\tilde{c}_{\text{OLH}}(i)] = n \cdot \frac{4e^\epsilon}{(e^\epsilon - 1)^2} \quad (5)$$

Eq.5 is identical to the variance of OUE. OLH achieves this accuracy while reducing the communication cost from $\Theta(d)$ to $\Theta(n)$ by transmitting a hash index. This mathematical equivalence proves that OLH reduces communication cost by orders of magnitude at no cost to utility.

4.2 Evaluation of Attacks on Frequency Estimation Techniques

We now analyze the effectiveness of the Maximal Gain Attack(MGA) against these protocols. We quantify security using the Expected Attack Gain(G), derived by Cao et al.[1] which measures the total frequency increase an attacker can force onto target items or sites.

Definition of Parameters: Before presenting the gains, we define the key parameters used in our analysis. Let n be the number of genuine users and m be the number of fake users injected by the attacker. We define $\beta = \frac{m}{n+m}$ as the fraction of fake users in the total population. The attacker aims to promote a set of target items T with size $r = |T|$. We denote the sum of the true frequencies of these target items among the genuine users as $f_T = \sum_{t \in T} f_t$. Finally, d represents the total domain size, and ϵ is the privacy budget.

4.2.1 Vulnerability of kRR

The attack against kRR is trivial(fake users report the target), but its impact is severe because the system implicitly trusts reports more when d is large (to compensate for high noise).

$$G_{\text{kRR}} = \beta(1 - f_T) + \frac{\beta(d - r)}{e^\epsilon - 1} \quad (6)$$

The term d in the numerator means the attack gain scales linearly with the domain size, making kRR significantly insecure for large vocabularies compared to OUE/OLH.

4.2.2 Vulnerability of OUE/OLH

Because of the constant gain in OUE and OLH, the attack is more sophisticated (using bit vectors or hash collisions), but the gain is theoretically stable. The approximate gain for both is:

$$G_{\text{OUE/OLH}} = \beta(2r - f_T) + \frac{2\beta r}{e^\epsilon - 1} \quad (7)$$

The gain depends on the number of target items r but is independent of d . While OUE and OLH are still vulnerable to poisoning, they are significantly more robust than kRR for large domains because the attack effectiveness does not explode as the dictionary size grows.

4.3 The Fundamental Security-Privacy Tradeoff

Connecting the utility and security analysis from Section 4.1 and Section 4.2, reveals a fundamental contradiction. In both gain equations Eq. 6 and Eq. 7, the dominant term contains the inverse of the privacy budget:

$$G_{\text{MGA}} \propto \frac{1}{e^\epsilon - 1}$$

This creates an less intuitive paradox that strengthening the privacy guarantee (lowering ϵ)mathematically necessitates increasing the noise ($p^* \rightarrow q^*$). This increased noise provides a larger statistical mask for fake users, allowing them to inject more bias without detection. Thus we derive the inverse relationship, implying that high privacy leads to high vulnerability.

5 Conclusions

Requires:

- conclusion
- open questions

References

- [1] X. Cao, J. Jia, and N. Z. Gong. Data poisoning attacks to local differential privacy protocols. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 947–964, 2021.
- [2] J. C. Duchi, M. I. Jordan, and M. J. Wainwright. Local privacy and statistical minimax rates. In *2013 IEEE 54th annual symposium on foundations of computer science*, pages 429–438. IEEE, 2013.
- [3] C. Dwork, A. Roth, et al. The algorithmic foundations of differential privacy. *Foundations and trends® in theoretical computer science*, 9(3–4):211–407, 2014.

- [4] Ú. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 1054–1067, 2014.
- [5] T. T. Nguyễn, X. Xiao, Y. Yang, S. C. Hui, H. Shin, and J. Shin. Collecting and analyzing data from smart device users with local differential privacy. *arXiv preprint arXiv:1606.05053*, 2016.
- [6] S. Wang, L. Huang, P. Wang, H. Deng, H. Xu, and W. Yang. Private weighted histogram aggregation in crowdsourcing. In *International Conference on Wireless Algorithms, Systems, and Applications*, pages 250–261. Springer, 2016.
- [7] T. Wang, J. Blocki, N. Li, and S. Jha. Locally differentially private protocols for frequency estimation. In *26th USENIX security symposium (USENIX Security 17)*, pages 729–745, 2017.

Appendix starts from here.