PrivSketch: A Private Sketch-based Frequency Estimation Protocol for Data Streams

Ying Li[1,2], Xiaodong Lee[1,2(✉)], Botao Peng[1(✉)], Themis Palpanas[3], and Jingan Xue[4]

[1] Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] LIPADE, Université Paris Cité French University Institute (IUF)
[4] Huawei Technologies

Abstract. Local differential privacy (LDP) has recently become a popular privacy-preserving data collection technique protecting users' privacy.The main problem of data stream collection under LDP is the poor utility due to multi-item collection from a very large domain. This paper proposes PrivSketch, a high-utility frequency estimation protocol taking advantage of sketches, suitable for private data stream collection. Combining the proposed background information and a decode-first collection-side workflow, PrivSketch improves the utility by reducing the errors introduced by the sketching algorithm and the privacy budget utilization when collecting multiple items. We analytically prove the superior accuracy and privacy characteristics of PrivSketch, and also evaluate them experimentally. Our evaluation, with several diverse synthetic and real datasets, demonstrates that PrivSketch is 1-3 orders of magnitude better than the competitors in terms of utility in both frequency estimation and frequent item estimation, while being up to ∼100x faster.

1 Introduction

Motivation. Collecting user data, often in the form of a data stream, in order to analyze them and provide some services has become a common practice. However, data collection may expose user information,which is a major concern. Local Differential Privacy (LDP) is popular to protect individual privacy during data collection and has been widely used in technology companies (such as Apple, Google, Microsoft). It perturbs data locally before sending them to the collector and enables the collector to obtain approximate statistics on the perturbed data, to avoid the risk of disclosing user privacy. A parameter ϵ is used to quantify the amount of perturbation, which determines the degree of privacy protection and the utility of the privacy-preserving algorithm.

Utility Problem. Although several studies have focused on the frequency estimation problem under LDP, they do not perform well when used in a data stream context, due to following reasons. First, existing solutions consider a unified size for the data items generated by different users (i.e., data length) that is based on unrealistic assumptions (assume only one item in a collection

interval) [10, 25], or a predefined/estimated unified size L for each collection (padding and sampling) [19, 22, 23], in both cases hurting utility. Second, the large domains of several data streams (e.g., URLs, and IP) lead to excessive computation and communication costs, as well as significant perturbation errors (some of the existing literature on frequency estimation [22, 23] is only applicable to small cardinality domains).

Sketching is widely used in streaming data processing for compressing sparse data from a large domain (e.g., when a tiny percentage of webpages are accessed by any individual user). The uniform size of sketches makes it possible to unify the data length of different users without extra padding and sampling [19], and leads to efficient storage. The sketching has been combined with LDP in the Private Count-Mean Sketch (PCMS) algorithm [20] proposed by Apple. However, it operates at the granularity of single items, which hurts performance. When considering extending it to multi-item collections, the following problems emerge. (i) The error introduced by sketching algorithms is not considered. Aggregating sketches from users directly is equivalent to encoding all data into one sketch, leading to increasing errors of collisions, i.e., data hashed in the same position. (ii) To maintain user-level privacy, allocating the privacy budget for each counter in the sketch is required, resulting in substantial inaccuracies and poor utility.

Our solution. We propose PrivSketch, a high-utility privacy-preserving sketch-based frequency estimation protocol that leads to lower errors when compared to existing solutions. PrivSketch proposes an innovative LDP collector-side workflow that decodes the perturbed sketch before aggregating and calibrating it, which avoids the error introduced by the collisions when aggregating all perturbed sketches in the traditional decode-after workflow. In addition, PrivSketch utilizes the ordering matrix extracted from the original sketch, which enables the collector to obtain the minimum index information, while ensuring the privacy of each user's sketch (cf. proof in Section 3.4). This effectively reduces the minimum calculation error caused by disturbance and is the first attempt to improve utility using background information. Furthermore, PrivSketch uses the sampling technique to improve the utilization of the information encoded by the sketch, transmitting relatively accurate information with a limited privacy budget. Thus, it reduces the error caused by the uniform allocation of the privacy budget when encoding multiple items.

Contributions. Our contributions are summarized below.

• We propose a novel LDP protocol, PrivSketch, that is suitable for frequency estimation in data streams where multi-item encoding is needed. It is the first sketch-based privacy-preserving protocol that considers the errors introduced by the sketches with a novel decode-first workflow. It employs background information to reduce the minimum value calculation errors of sketches and utilizes a sampling technique to improve the privacy budget utilization.

• We prove (cf. Section 3.4) that the ordering matrix as background information does not expose the original value of each counter in the sketch, which meets the privacy needs of users. We introduce a new definition of the indistinguishable input set, where the collector cannot distinguish any two values.We

observe that the utility of LDP algorithms can be improved with appropriate additional background information , but does not harm the users' privacy.

• We evaluate our approach on both synthetic and real datasets. We compare it with extensions and variants of existing algorithms, including the multi-item encoding extension and its min-variant algorithm. The utility of the protocol proposed in this paper is 1-3 orders of magnitude more accurate than existing algorithms, and up to ∼100x faster.

2 Background and preliminaries

Local Differential Privacy (LDP). Differential privacy (DP) [13] is a technology with quantified privacy protection, but relies on a trustworthy third-party collector. To remove the trust in the collector, LDP [11] was proposed where original data are only accessible by users, and the collector only receives the perturbed data. A mechanism $\mathcal{M}$ satisfying LDP can be defined as follows.

Definition 1 (ϵ-Local Differential Privacy [11]). *A randomized algorithm $\mathcal{M}$ satisfies ϵ-local differential privacy ($\epsilon > 0$), if and only if for any two input tuples $x, x' \in \mathcal{D}$ and output y, then $\frac{\Pr[\mathcal{M}(x)=y]}{\Pr[\mathcal{M}(x')=y]} \leq e^{\epsilon}$.*

Thus, a smaller ϵ means large perturbation and more indistinguishable, but lower utility. There is an important property of LDP:

Theorem 1 (Sequential Composition Mechanism [17]). *Assume a randomized algorithm $\mathcal{M}$ consists of a sequence of randomized algorithms $\mathcal{M}_i (1 \leq i \leq t)$. When for each $i, \mathcal{M}_i$ satisfies ϵ_i-LDP, $\mathcal{M}$ satisfies $\sum_{i=1}^{t} \epsilon_i$-LDP.*

Randomized Response Mechanism (RR) [28, 15]. This fundamental LDP mechanism achieves plausible deniability by allowing users not to give the original value. Specifically, for binary values, users answer the original value with probability p, and the opposite value with probability $q = 1 - p$. To achieve ϵ-LDP, the worst case is $\frac{\max \Pr[\mathcal{M}(x)=y]}{\min \Pr[\mathcal{M}(x')=y]} = \frac{p}{1-p} = e^{\epsilon}$, therefore $p = \frac{e^{\epsilon}}{1+e^{\epsilon}}$. Denote $\Pr[x=1]$ the percentage of $x=1$. For the collector, $\Pr[y=1] = p\Pr[x=1] + (1-p)(1-\Pr[x=1])$ and $\Pr[y=0] = p(1-\Pr[x=1]) + (1-p)\Pr[x=1]$. $\Pr[y=1]$ and $\Pr[y=0]$ represent the probability of the output y taking the value of 1 and 0, respectively, which can used to obtain the unbiased estimation of $\Pr[x=1]$ and $\Pr[x=0]$.

Count-Min Sketch (CMS). A common approach to compress data from a large domain is the sketching algorithm. The Count-Min Sketch [9] is one of the most popular sketching algorithms due to its efficiency. The sketching uses a matrix X consisting of $K \times M$ counters, bound to K hash functions $H_1, H_2, \ldots, H_K : \{1, \ldots, d\} \mapsto \{1, \ldots, M\}$. It consists of two phases: (i) update, where K hash functions are used to hash the updated item x, and then the corresponding counters are updated, i.e. $X_{k,H_k(x)} = X_{k,H_k(x)} + 1, \forall 1 \leq k \leq K$; (ii) query, where item x's count $c(x)$ is estimated, denoted by $\widetilde{c}(x)$, based on the corresponding counters in the sketch, i.e. $\min_{1 \leq k \leq K} X_{k,H_k(x)}$ [9].

Private Count-Mean Sketch (PCMS-Mean) [20]. PCMS-Mean estimates frequency under LDP, where the user perturbs data before sending them to the collector. Specifically, for item x, each user chooses a hash function H_k and updates $X_{k,H_k(x)} = 1$ (other positions keep as -1), then, perturbs X_k using RR and sends the perturbed result $\hat{X}_k$ to the collector. The collector constructs a matrix of size $K \times M$ where each row is the sum of the perturbed rows indexed by k, and estimates the frequency by averaging the sum of k counters corresponding to K hash functions. The algorithm assumes that each user generates only one item. Thus, for any two rows from different users X_k and $X'_{k'}$, at most two positions can be different. To protect these two positions under privacy budget ϵ, the parameter p in RR is set to $\frac{e^{\epsilon/2}}{1+e^{\epsilon/2}}$ (cf. Theorem 1).

When extending PCMS-Mean to encode multiple items, the number of different positions in any two rows from different users is up to M due to unlimited items of each user. Thus, to protect the privacy of each position, the parameter p is set to $\frac{e^{\epsilon/M}}{1+e^{\epsilon/M}}$. This naive solution works poorly when M is large. The irrational allocation of ϵ is one of the reasons. In addition, the error introduced by the sketching algorithm is also non-negligible. The estimation error of different sketching algorithms varies. The error of the Count-Min Sketch is smaller than that of the Count-Mean Sketch [8]; hence, we use the Count-Min Sketch.

Problem Definition This paper studies the frequency estimation problem under LDP for data streams, where data are generated from a very large domain. There is an untrusted collector and a set of n users represented by $U = \{U_1, U_2, \ldots, U_n\}$. Each user, U_i, has a set of items of length $L^{(i)}(L^{(i)} \geq 0)$, which is denoted by $S^{(i)} = \{S_1^{(i)}, S_2^{(i)}, \ldots\}, |S^{(i)}| = L^{(i)}$. Each item $S_\ell^{(i)}(0 \leq \ell \leq L^{(i)})$ is discrete value and drawn from a large domain $\mathcal{D}$ of size $|\mathcal{D}| = d$, that is, $S_\ell^{(i)} \in \mathcal{D}$. In this paper, we focus on estimating the frequency of each item from $\mathcal{D}$, that represents the proportion of users who possess the item. Formally, the frequency for each value $x \in \mathcal{D}$ is defined as: $f(x) = \frac{|\{i|\exists \ell, 0 \leq \ell \leq L^{(i)}, S_\ell^{(i)} = x\}|}{n}$.

3 PrivSketch Solution

PrivSketch is a LDP protocol based on CMS to solve the frequency estimation problem in data stream collection. PrivSketch uses a novel collector-side workflow (cf. Section 3.2) and the ordering matrix (cf. Section 3.2) to reduce errors introduced by sketches. PrivSketch also uses a sampling technique to increase the information utilization in sketches under a limited privacy budget.

Fig. 1 provides a high-level overview of PrivSketch workflow. At the user end, the encoder encodes items using CMS and the perturber perturbs a sampled one counter in the sketch using RR. Then, the perturbed counter $\hat{X}_{k,m}^{(i)}$ is sent to the collector with an ordering matrix $O^{(i)}$ which reflects the order of all counters in the original sketch $X^{(i)}$. At the collector end, the decoder restores $\hat{X}_{k,m}^{(i)}$ to the original domain $\mathcal{D}$ by calculating each item's minimum index based on $O^{(i)}$ and updating counts of items x whose minimum index equal to the sampled k

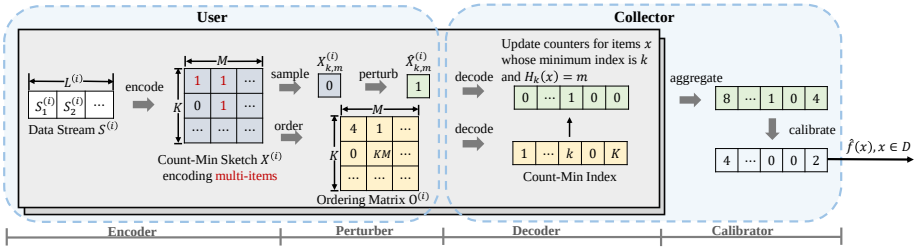


Fig. 1. Overview of PrivSketch.

Algorithm 1: PrivSketch

Input: $\{S^{(1)}, S^{(2)}, \ldots, S^{(n)}\}, \epsilon, K, M, D \subset \mathcal{D}$

select a set of hash functions $\mathcal{H} = \{H_1, H_2, \ldots, H_K\}$;
for *each* $i \in [1, n]$ **do**
  $\hat{X}^{(i)}, O^{(i)} \leftarrow$ PrivSketch-User $(S^{(i)}, \epsilon, n, K, M, \mathcal{H})$;
  send $\hat{X}^{(i)}, O^{(i)}$ to the collector;
for *each* $x \in \mathcal{D}$ **do**
  set $\hat{\mathcal{X}} \leftarrow \{\hat{X}^{(1)}, \hat{X}^{(2)}, \ldots, \hat{X}^{(n)}\}$;
  set $\mathcal{O} \leftarrow \{O^{(1)}, O^{(2)}, \ldots, O^{(n)}\}$;
  $\hat{f}(x) \leftarrow$ PrivSketch-Collector $(x, \epsilon, n, M, \mathcal{H}, \hat{\mathcal{X}}, \mathcal{O})$;
return $\{\hat{f}(x) | x \in D\}$

and $H_k(x) = m$. Then, the calibrator estimates items' frequency by aggregating restored counts from users and calibrating the perturbation error. The protocol is shown in Algorithm 1. We elaborate on its novel designs and details next.

3.1 Decoding-First Collector-Side Workflow

An important characteristic of PrivSketch is the decoding-first feature on the collector side, which is designed to reduce the collisions in the private sketching algorithm. The naive protocol, PCMS-Min as traditional LDP protocols, consists of three steps: Encode, Perturb, and Aggregate [24]. Collisions can occur in the Encode and Aggregate procedure. During encoding, the collision is caused by that different items are hashed into the same positions, which can be reduced by a good choice of the sketching parameters. During aggregation, sketches from n users are integrated into one sketch, equivalent to encoding data from n users using the same sketch. This leads to a high probability of collisions due to the large number of users under LDP. We find that decoding the perturbed data before aggregation can avoid this collision, where the Decode procedure has been implemented by the collector after the Aggregate procedure but ignored by LDP protocol designers. If the collector decodes the perturbed data before aggregation, only the perturbed counts instead of the sketches are aggregated, thus, no collisions. We present theoretical proof for how the decode-first workflow reduces collision errors following. Note in our design, we use Calibration instead

of Aggregate to describe the procedure where aggregating and calibrating errors caused by the perturbation.

Theorem 2. *For estimating the frequency of a value $x \in \mathcal{D}$ using Count-Min Sketch,* $\min_k \sum_{i=1}^{n} X^{(i)}_{k,H_k(x)}$ *represents the results of aggregating sketches before decoding,* $\sum_{i=1}^{n} \min_k X^{(i)}_{k,H_k(x)}$ *represents the results of decoding sketches before aggregating, the following formula holds:*

$$\min_k \sum_{i=1}^{n} X^{(i)}_{k,H_k(x)} \geq \sum_{i=1}^{n} \min_k X^{(i)}_{k,H_k(x)} \geq nf(x) \tag{1}$$

where $f(x)$ represents the true frequency of x.

Proof. For each user U_i and any $1 \leq k \leq K$, $X^{(i)}_{k,H_k(x)}$ reflects the occurrence of both x and $x'(x' \neq x)$, which are hashed into the same position with x.

$$X^{(i)}_{k,H_k(x)} = \mathbb{1}\{x \in S^{(i)}\} \vee \mathbb{1}\{x' \in S^{(i)}, H_k(x) = H_k(x')\}.$$

For the minimum index k where $X^{(i)}_{k,H_k(x)}$ is minimal, the equation above holds. As a result, $\sum_{i=1}^{n} \min_k X^{(i)}_{k,H_k(x)} = nf(x) + \sum_{i=1}^{n} \mathbb{1}\{x' \in S^{(i)}, x \notin S^{(i)}, H_{\min_k}(x) = H_{\min_k}(x')\} \geq nf(x)$. Moreover, $\sum_{i=1}^{n} X^{(i)}_{k,H_k(x)} \geq \sum_{i=1}^{n} \min_k X^{(i)}_{k,H_k(x)}$, $1 \leq k \leq K$. Considering $\min_k$ is one of the case that belongs to $[1, K]$, we can conclude that $\min_k \sum_{i=1}^{n} X^{(i)}_{k,H_k(x)} \geq \sum_{i=1}^{n} \min_k X^{(i)}_{k,H_k(x)}$.

Thus, when an unbiased estimation of the query result of the original Count-Min Sketch is achieved, the decode-first collector-side workflow brings fewer errors. Next, we introduce how to ensure an unbiased estimation in PrivSketch.

3.2 Ordering Matrix Generation

In PrivSketch, the minimum index of the perturbed count can be changed by the randomized response mechanism which hinders an unbiased estimation. As shown in Fig. 1, the collector queries the perturbed sketch $\hat{X}^{(i)}$ and estimates based on it. Assume the calibration for estimation of the frequency $f(x)$ in $\mathcal{D}$, is based on sketches $\hat{X}^{(i)}$ with a linear function $h(x)$, i.e. $\hat{f}(x) = h(\sum_{i=1}^{n} \min_k \hat{X}^{(i)}_{k,H_k(x)})$. PrivSketch needs to satisfy the expectation of the variable after perturbation is an unbiased estimation of the result from querying the original sketch $\widetilde{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \min_k X^{(i)}_{k,H_k(x)}$. That is,

$$\mathbb{E}[\hat{f}(x)] = \mathbb{E}[h(\sum_{i=1}^{n} \min_k \hat{X}^{(i)}_{k,H_k(x)})] = \widetilde{f}(x) = \frac{1}{n} \sum_{i=1}^{n} \min_k X^{(i)}_{k,H_k(x)}.$$

Assuming that the row indices of the minimum count for x in the perturbed and original sketches are k' and k, if $k \neq k'$,

$$\mathbb{E}[\hat{X}^{(i)}_{k',H'_k(x)}] = pX^{(i)}_{k',H'_k(x)} - qX^{(i)}_{k',H'_k(x)} = (p-q)X^{(i)}_{k',H'_k(x)} \geq (p-q) \min_k X^{(i)}_{k,H_k(x)},$$

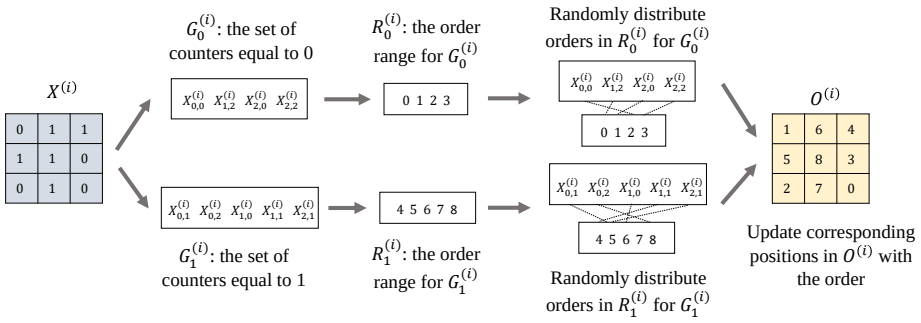


Fig. 2. The process of generating the ordering matrix.

where p and q represent the probability of keeping the original value and flipping to the opposite value, respectively. Due to the randomization, the minimum count in the perturbed sketch is not always in the same position as in the original sketch, i.e., $k \neq k'$. However, because the gap between different counts in sketches is diverse and related to the count of specific items, it is difficult to turn the above inequality into an equation by constructing a $h(x)$. To solve this problem, we propose the ordering matrix.

The ordering matrix $O^{(i)}$ is the background information provided by users, to assist the collector in getting the same row index of the minimum value as the original matrix, which takes advantage of the insensitivity of LDP to any background information to keep the privacy. The ordering matrix $O^{(i)}$ is a $K \times M$ matrix, where each position represents the serial number of the corresponding position in the original sketch $X^{(i)}$ ordered by count. Firstly, each counter $X_{k,m}^{(i)}$ is distributed into different groups $G_v^{(i)}$ according to its count v. As a result, $G_v^{(i)}$ includes a set of counters $\{(k,m)|X_{k,m}^{(i)} = v\}$ and its length is denoted by $|G_v^{(i)}| = g_v$. Secondly, each group G_v is bound with its order range $R_v^{(i)} = [\sum_{v' \leq v} g_{v'}, \sum_{v' \leq v} g_{v'} + g_v]$. Thirdly, we randomly sample an order without replacement from $R_v^{(i)}$ for each counter in $G_v^{(i)}$ where the order selected for each counter $X_{k,m}^{(i)}$ is denoted as $r_{k,m}^{(i)}$. Finally, we update the ordering matrix $O_{k,m}^{(i)} = r_{k,m}^{(i)}$. Thus, the collector can get the same minimum index by comparing the order of counters in $O^{(i)}$: this has the same result as calculating the minimum index on the original sketch $X^{(i)}$. An example is shown in Fig. 2.

In the following, we prove the estimation is unbiased in PrivSketch (Section 3.3), and analyze the impact of the ordering matrix on privacy (Section 3.4).

3.3 Utility Proof and Improvements

We present the protocol details on the user- and collector-side. We prove that the estimations are unbiased, and analyze the variances of errors, then employ sampling to achieve high utility.

Algorithm 2: PrivSketch-User

Input: $S^{(i)}, \epsilon, n, K, M, \mathcal{H}$

1 initialize a sketch $X^{(i)} \leftarrow \{0\}^{K \times M}$;
2 **for** *each* $\ell \in [1, L^{(i)}]$,*each* $k \in [1, K]$ **do**
3 $\quad X^{(i)}_{k, H_k(s^{(i)}_\ell)} = 1$;
4 generate the ordering matrix $O^{(i)}$;
5 **for** *each* $k \in [1, K]$, *each* $m \in [1, M]$ **do**
6 $\quad$ sample r from $[0, 1]$ uniformly;
7 $\quad$ **if** $r < \frac{1}{e^{\epsilon/KM}+1}$ **then**
8 $\quad\quad \hat{X}^{(i)}_{k, H_k(s^{(i)}_\ell)} = -2X^{(i)}_{k, H_k(s^{(i)}_\ell)} + 1$;
9 $\quad$ **else**
10 $\quad\quad \hat{X}^{(i)}_{k, H_k(s^{(i)}_\ell)} = 2X^{(i)}_{k, H_k(s^{(i)}_\ell)} - 1$;
11 **return** $\hat{X}^{(i)}, O^{(i)}$

Algorithm 3: PrivSketch-Collector

Input: $x, \epsilon, n, M, \mathcal{H}, \hat{\mathcal{X}}, \mathcal{O}$

1 select a set of hash functions $\mathcal{H} = \{H_1, H_2, \ldots, H_K\}$;
2 $C(x) \leftarrow 0$;
3 **for** *each* $i \in [1, n]$ **do**
4 $\quad k_{\min} \leftarrow \arg\min_k O^{(i)}_{k, H_k(x)}$;
5 $\quad C(x) \leftarrow C(x) + \hat{X}^{(i)}_{k_{\min}, H_{k_{\min}}(x)}$;
6 $\hat{f}(x) \leftarrow \frac{1}{2}(\frac{e^{\epsilon/KM}+1}{e^{\epsilon/KM}-1}\frac{C(x)}{n} + 1)$;
7 **return** $\hat{f}(x)$

User-side protocol (Algorithm 2). It consists of an encoder (lines 1-4) and a perturber (lines 5-10). In the encoder, each user records locally whether x appears, because our objective is to obtain the frequency of any value x in $\mathcal{D}$ (instead of counts). Consequently, an update in the encoder is a boolean disjunction, not an integer addition. Each position $X_{k,m}$ is initialized as *False* (i.e., 0). When x is hashed to $X_{k,m}$, the update is $X_{k,m} = X_{k,m} \vee True = True$ (line 3). After encoding, the ordering matrix is computed by the encoder. The perturber uses the randomized response mechanism (as in PCMS-Mean) to perturb each value to the opposite value with a probability of $\frac{1}{e^{\epsilon/KM}+1}$ due to at most $K \times M$ different positions.

Collector-side Protocol (Algorithm 3). First, the decoder estimates the perturbed frequency of the value x using the perturbed minimum in $\hat{\mathcal{X}}$. The position of the minimum is provided by the background information $\mathcal{O}$ (line 4). Next, the calibrator removes the perturbation error to obtain the final estimation (line 6). The utility proof of the protocols follows.

Theorem 3. *Let $C(x)$ denote the perturbed counters for each value in $\mathcal{D}$. $\hat{f}(x) = \frac{1}{2}(\frac{e^{\epsilon/KM}+1}{e^{\epsilon/KM}-1}\frac{C(x)}{n}+1)$ is an unbiased estimation of $\widetilde{f}(x) = \frac{1}{n}\sum_{i=1}^{n}\min_k X^{(i)}_{k,H_k(x)}$ which is the frequency inferred from the original count-min sketch. Furthermore, the variance of $\hat{f}(x)$ is $\frac{e^{\epsilon/KM}}{n(e^{\epsilon/KM}-1)^2}$.*

Proof. For each user U_i, the counters for the item x in row k of perturbed sketch $\hat{X}$ is denoted by $\hat{X}^{(i)}_{k,H_k(x)}$, which value is determined by $X^{(i)}_{k,H_k(x)}$ (lines 6-10 in Algorithm 2). $C(x)$, which represents the result by aggregating the perturbed counters at the minimum position $\min_k \hat{X}^{(i)}_{k,H_k(x)}(x)$, equal to $\sum_{i=1}^{n}\min_k \hat{X}^{(i)}_{k,H_k(x)}(x)$, satisfies: $\mathbb{E}[C(x)] = 2(p–q)n\widetilde{f}(x)+(2q–1)n$, $\mathrm{Var}[C(x)] = 4n\{(p+q-1)(p-q)\widetilde{f}(x)+q(1-q)\}$, where $n\widetilde{f}(x)$ is the estimated number of users with x in their sequences using the set of original sketch $\mathcal{X}$. In our protocol, $p = \frac{e^{\epsilon/KM}}{e^{\epsilon/KM}+1}, q = \frac{1}{e^{\epsilon/KM}+1}$. Thus, the expectation of $\hat{f}(x)$, can be shown to be equal to $\widetilde{f}(x)$ as follows, which means the estimation is unbiased. And its variance of $\hat{f}(x)$ is satisfied:

$$\mathbb{E}[\hat{f}(x)] = \frac{1}{2}(\frac{e^{\epsilon/KM}+1}{e^{\epsilon/KM}-1}\frac{\mathbb{E}[C(x)]}{n}+1) = \widetilde{f}(x) \tag{2}$$

$$\mathrm{Var}[\hat{f}(x)] = \frac{1}{4n^2}\frac{(e^{\epsilon/KM}+1)^2}{(e^{\epsilon/KM}-1)^2}\mathrm{Var}[C(x)] = \frac{e^{\epsilon/KM}}{n(e^{\epsilon/KM}-1)^2}. \tag{3}$$

Sample the sketches. Following the above design, larger K and M make the perturbation probability closer to $\frac{1}{2}$ as random.And the variance also increases at the same time. The limited privacy budget ϵ/KM for each counter makes the collector receive scarcely useful information from the perturbed sketches, making it difficult to infer the true frequency. To solve the problem, the sampling technique is a common solution, i.e., randomly sampling one from $K \cdot M$ counters on the user end. Thus, for each counter chosen, the privacy budget becomes ϵ. The variance now is $\mathrm{Var}[\hat{f}(x)] = \frac{KMe^{\epsilon}}{n(e^{\epsilon}-1)^2}$, which is linearly related to $K \cdot M$ due to the sampling error, thus increasing more slowly than the exponential relation in Equation (3). However, it is challenging to obtain the optimal sketching, because as K and M increase, the collision error introduced by Count-Min Sketch decreases, which is also related to the data domain size d and its distribution [9].Though, we experimentally evaluate the effect of K and M on frequency estimation in Section 4.2. Besides, the utility of sampling in sketches is also verified by comparing with traditional PSFO [26] in Section 4.1.

3.4 Privacy Analysis

When the user sends only the perturbed counter $\hat{X}^{(i)}_{k,m}$ to the collector with the flipping probability $\frac{1}{e^{\epsilon}+1}$, ϵ-LDP is satisfied. However in PrivSketch, the user need also send the ordering matrix $O^{(i)}$ to the collector which may expose useful messages and indirectly damage the privacy. In the following, we analyze the influence of $O^{(i)}$ on privacy.

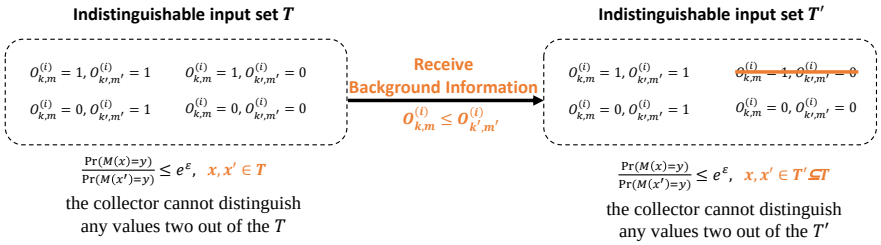


Fig. 3. Example of the effect of background information on indistinguishable input set.

The ordering matrix $O^{(i)}$ can be utilized to exclude some possible inputs for the collector, but the collector still cannot distinguish some inputs. As Fig. 3 shows, if $O^{(i)}_{k,m} \le O^{(i)}_{k',m'}$, $X^{(i)}_{k,m} = 1$ and $X^{(i)}_{k',m'} = 0$ will not hold at the same time. Thus, the cases of the possible sketches of users are reduced from 4 to 3 in the collector's view. To quantify the effect of the background information, we introduce *indistinguishable input set* to represent the possible inputs in the collector's view, denoted by T. According to the LDP definition, any two inputs are indistinguishable regardless of any background knowledge from the adversary. Therefore, we can deduce that any two inputs in the indistinguishable set still satisfy the LDP definition, even though the indistinguishable set becomes smaller than without the background information.

Theorem 4. *Consider a mechanism $\mathcal{M}$ that satisfies ϵ-LDP, its indistinguishable input set T, and any two inputs x, x'. When the collector receives any output y, along with the background information I, there exists an indistinguishable set $T' \subseteq T$ satisfying the following inequality:* $\frac{\Pr[M(x)=y]}{\Pr[M(x')=y]} \le e^{\epsilon}, \quad x, x' \in T'$.

Proof. For any I, T can be divided into two parts, T_+ and T_-. The former represents the inputs that are consistent with the information I, i.e., the possible inputs when I is true. The latter includes the inputs that contradict the information I, that is, the impossible inputs when I is true. Based on I, the collector can infer that the original input belongs to $T_+ (\subseteq T)$. For any two inputs $x, x' \in T_+$, x, x' is also in T. Therefore, following the definition of ϵ-LDP, $\frac{\Pr[M(x)=y]}{\Pr[M(x')=y]} \le e^{\epsilon}$ is satisfied and any two input $x, x' \in T'$ is distinguishable.

The indistinguishable input set T' computed by the ordering matrix $\mathcal{O}$ is enough to protect the privacy of users in our problem. In PrivSketch, what each user needs to protect is its original sketch matrix $X^{(i)}$. Thus, the collector should not infer the value of any counter in $X^{(i)}$ is 1 or 0. In PrivSketch, counters can be divided into two groups, G_1 and G_0, and $g_1 + g_0 = KM$. Thus, when the collector receives $O^{(i)}$, the indistinguishable input set T' at most includes $KM + 1$ possible sketches with different sizes of each group. There are some constraints for sketches, e.g., it is impossible that $g_1 = 1, 2, 3$, because when there is an item occurred, for each $k \in [1, K]$, $\exists (k, m) \in G_1, m \in [1, M]$. Nevertheless, $\{0\}^{KM}, \{1\}^{KM} \in T'$ always holds. Thus, there is no counter with

Table 1. Datasets characteristics

Dataset	n	d	max	min	P_{90}	**Dataset**	n	d	max	min	P_{90}
Kosarak	990002	41270	2498	1	15	AOL	521693	1632788	61932	1	62
Dataset1	10000	100000	123	1	80	Dataset2	100000	100000	117	1	78
Dataset3	100000	20000	112	1	73	Dataset4	100000	40000	107	1	72
Dataset5	100000	60000	110	1	74	Dataset6	100000	80000	109	1	75

same value in different possible inputs, that is, its value is equal to 1 in some inputs and equal to 0 in the other inputs. The collector still cannot determine the value of each counter, which is sufficient to protect the privacy of users.

4 Experimental Evaluation

In this section, we evaluate the utility and running time of PrivSketch over synthetic and real datasets, and analyze how the main parameters affect its performance. For a comprehensive evaluation, we compare PrivSketch to the state-of-the-art PCMS-Mean [20], and PSFO [26] based on OLH [24] (denoted as PS-OLH in our experiments) for frequency estimation, and SVIM [26], a two-phase heavy hitter discovery protocol for discovery of frequent items.

Environment. We implement all LDP protocols in Python and conduct experiments on a server with 2 Intel Xeon 3206R Processors and 32G RAM running Centos. We repeat each experiment 10 times and report the average results.

Datasets. We use 6 synthetic datasets and 2 real datasets (see Table1).

• **Synthetic Datasets**: These datasets follow Zipf distribution that real data stream often conforms to, with different number of users n and domain size d.

• **Kosarak** [4]: This dataset contains the clicked items that anonymized users from a Hungarian online news portal, involving nearly 1M users and 40K items.

• **AOL** [12]: This dataset contains search queries of users on AOL between March 1 and May 31, 2016, with corresponding URLs clicked by them. The dataset includes more than 500K users with 1.6 million distinct URLs.

Parameters. The number of hash functions K is set to 4, and each hash function's hash domain size M is set to 128. The default privacy budget ϵ is 3, within the acceptable range in many works [8, 27, 19].

Evaluation Measures. We use the following measures, including running time.

• **Mean Squared Error (MSE)**. We evaluate the frequency estimation accuracy by MSE: $\frac{1}{d}\sum_{x\in\mathcal{D}}(\hat{f}(x)-f(x))^2$, where $f(x)$ is x's true frequency.

• **Variance (Var)**. We measure the error of estimating the top-k frequency terms using variance: $\frac{1}{|C_e\cap C_t|}\sum_{x\in C_e\cap C_t}(n\hat{f}(x)-nf(x))^2$.

• **Normalized Cumulative Rank (NCR)**. To evaluate the estimation of frequent items, NCR measures how many top-k items are identified by the protocol with a quality function $q(.)$. It is calculated as follows: $\sum_{x\in C_e} q(x)/\sum_{x'\in C_t} q(x')$, where C_t and C_e represents the true top-k items and the estimated top-k items respectively. For $x \in C_t$ with a rank i, $q(x) = k + 1 - i$. For $x \notin C_t$, $q(x) = 0$.

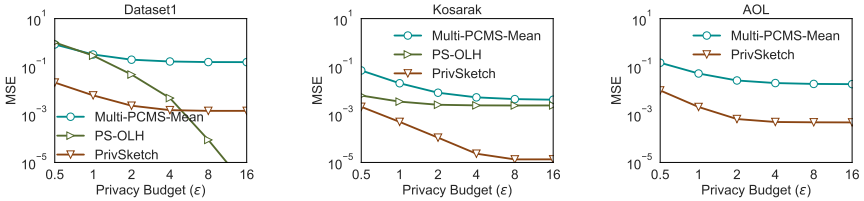


Fig. 4. Experimental results for frequency estimations.

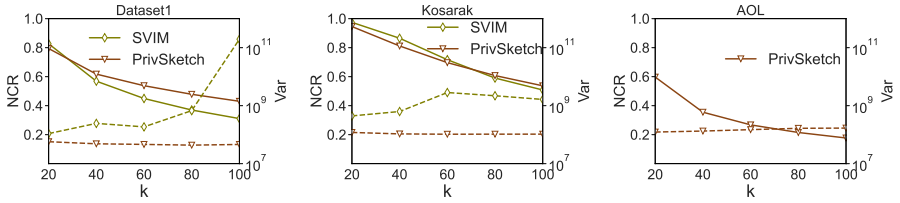


Fig. 5. VAR and NCR when varying parameter k.

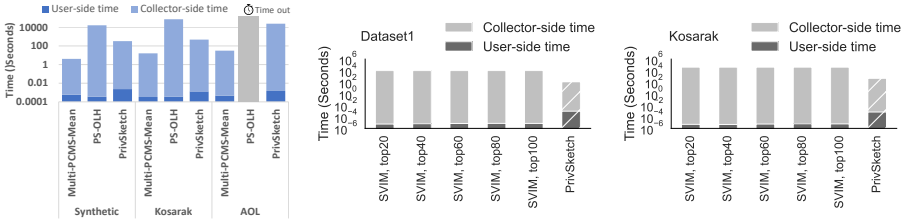
4.1 Comparing to Advanced Protocols

Experiments for Frequency Estimation: We compare our protocol to two advanced solutions: (i) a sketch-based solution, Multi-PCMS-Mean, which is an extended version of PCMS-Mean [20] for multi-item collection, and (ii) a non-sketch-based solution, PS-OLH, which is an advanced PSFO [26]. PSFO [26] combines the padding and sampling technique with a basic frequency estimation protocol to transform multiple-item into one-item problems. Because the optimal local hash (OLH) [26] performs best when $d \geq 3e^{\epsilon} + 2$ (i.e., for large domains), we choose the PSFO with OLH, i.e., PS-OLH, as our competitor. For a fair comparison, we assume the distribution of user input length is known and set the padding length l of PS-OLH to the 90th percentile of the user input [19] (avoiding to use the privacy budget to estimate l).

We evaluate the MSE of frequency estimation under different privacy budgets, varying from 0.5 to 16, on synthetic and real datasets. As shown in Fig. 4, PrivSketch performs best, especially for small privacy budgets, which indicates the high utility of PrivSketch and its strong privacy protection.

Experiments for Frequency Item Estimations: We also evaluate the performance of PrivSketch in frequent item mining (i.e., heavy hitter discovery), a popular application of frequency estimation. We compare it with the existing advanced multi-phase protocol, SVIM [26], which is the improved work after LDPMiner [19] and is also applicable in large domains. As shown in Fig. 5, PrivSketch performs better than SVIM, especially in frequency estimation for top-k items. It is expectable because PrivSketch has been designed for accurate frequency estimation, not frequent item identification.

Evaluation of running time. As shown in Fig. 6, Privsketch maintains a user-side running time smaller than $0.01s$ while performing the calculation of the ordering matrix. Overall, PrivSketch is faster than PS-OLH and SVIM about 100



(a) Frequency estimation (b) Frequent item estimation

Fig. 6. Comparison of running times.

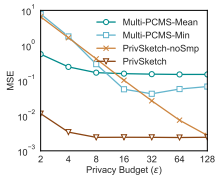
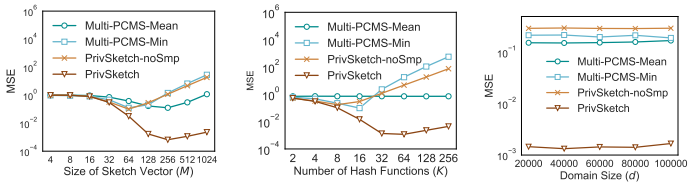


Fig. 7. MSE when varying ϵ, $n = 10^4$.



(a) vary M (K=4) (b) vary K (M=32) (c) vary d

Fig. 8. MSE when varying parameters K, M, d.

times, but slower than Multi-PCMS-Mean with much larger MSEs (in Fig. 4). The long running time of PS-OLH, SVIM and PrivSketch is the sacrificed time of reducing domain cardinality to gain high utility. Thus they need to restore the estimated items to the original domain for each user on the collector side, resulting in a complexity of $\mathcal{O}(nd)$. In PrivSketch, each user shares the same hash functions instead of local hash functions used in PS-OLH and SVIM, resulting in fewer hash function calculations on the collector side. We omit experimental results for PS-OLH and SVIM over AOL, because they need more than 10 days to compute, making them cumbersome to use in practice.

4.2 Experiments with Different Parameters

In this section, we compare PrivSketch with other sketch-based solutions to present the effect of our design under different parameters. In addition to Multi-PCMS-Mean, its min-estimation variant (denoted by Multi-PCMS-Min) and a middle version of PrivSketch without sampling (denoted by PrivSketch-noSmp) are also compared, to show the better utility of min estimation and the effect of our decode-first and sampling design.

Utility with small number of users. We evaluate the MSE on Dataset1 with 10^4 users under a privacy budget range $[2, 128]$. Note the unrealistic privacy budget used here is to show the effect of our designs. In Fig. 7, PrivSketch always performs best especially under a small ϵ. We observe similar results (omitted for brevity) when n varies in $[10^4,10^6]$. The result verifies that decode-first workflow with the ordering matrix effectively reduces the collision probability of sketches, and the min estimation has better accuracy than the mean estimation.

Impact of the size of the domain. We conduct this experiment on a group of synthetic datasets, which sets K, M and ϵ with default values, fixes $n = 10^5$, and varies the domain size d. As shown in Fig. 8(c), the errors of the four protocols only slightly increase with the increase of d. Theoretically, in a larger domain, when the sketch size is fixed, the collision probability increases, leading to an increase in error. However, since the items held by each user are sparse compared to the domain space, and the distribution of the number of items held by each user changes a little, the domain size change has a small impact. This confirms that sketching is an effective domain reduction and encoding method for data collection from a large domain.

Impact of the parameters of the sketch. In Fig. 8, we evaluate the effects of different K and M of the sketching using the datasets with parameters $n = 10^5$, $d = 10^5$ under $\epsilon = 3$. In Fig. 8(a), we can see that the utility of the PrivSketch is far better than the other three protocols under different M while fixing the hash vector size $K = 4$. As expected, increasing the size of the hash vector can reduce the estimation error. However, when M increases to a specific value, the error does not decrease but increases. This is because M affects two types of errors in these sketch-based LDP protocols. When M increases, the collision probability decreases, but the perturbation probability or the sampling errors increases. Varying the K brings a similar result to M, as shown in Fig. 8(b). However, the effects of K on Multi-PCMS-Mean protocol is different. Changes of K do not affect its MSE, because the effect of choosing one of the K hash functions when encoding is eliminated by the sum of K counters corresponding to K hash functions during the estimation process.

5 Related Work

Set-valued Data Collection. The diverse set size is a challenge for set-value data collection under LDP. Padding and Sampling [19] is a common way to unify the set length, such as in PSFO [26], PrivSet [22]. Although Wang [23] proposes the wheel mechanism to reduce the computational overhead, these works do not aim at a large domain, where an efficient data structure is needed. Many works [19, 26, 3, 27], focus on frequent item mining, also known as heavy-hitters discovery in a huge domain. They utilize a multi-phase strategy to reduce the large domain size first, using a small part of the privacy budget to discover frequent candidates, and using the remaining part to obtain an accurate estimation. Nevertheless, this strategy is not suitable for estimating frequency.

Frequency estimation with Hash-Encoding Technique. Under LDP settings, to reduce the data domain, RAPPOR [15] adopts Bloom filters to encode data, which requires expensive computations to use LASSO regression for the estimation. OLH [24] utilizes local hash functions to encode the user data, which requires a large number of hash calculations. With a simpler estimation solution, Count-Mean Sketch [20] was proposed to compute the populated emoji in IOS. [21, 18] improve it by sending multiple sketches for each user, which also brings extra communication costs. [3] uses Count-Median Sketch with the Hadamard

transform when computing the heavy hitters. [8] analyzes and compares LDP protocols with different sketching algorithms, including the Count-Min Sketch. However, these protocols, designed for the one-item collection, do not consider the error introduced by the sketching algorithm. Recently, [30] utilized hash functions to compute the frequency and the mean estimation of the k-sparse vector, with an assumption on the number of items each user generates.

Variants of LDP. Lots of works focus on optimizing the variants of LDP to improve its utility. Some works introduce extra trust in LDP, such as shuffling anonymized reports from users [7, 14], and combining the centralized DP with the local version [2]. Some works introduce an extra parameter to relax the privacy constraint, such as [1, 16] that use the distance metric of two inputs to improve the utility, which is inspired by the geo-indistinguishability concept [5]. Finally, some studies propose discriminative LDP based on different aspects, such as personalized privacy demand [6, 29]. These works do not utilize the background information to enhance utility as we do in this paper.

6 Conclusions

This paper studies the frequency estimation problem under local differential privacy. We propose a privacy-preserving data collection protocol, PrivSketch, which does not expose the original value of any counter in the sketch. We experimentally verify the effectiveness of PrivSketch: it outperforms existing LDP protocols by 1-3 orders of magnitude and executes up to ∼100x faster.

Acknowledgments We sincerely thank Dr. Zhenyu Liao for his insightful and constructive comments and suggestions on mathematical proof that help to improve the quality. This work is funded by NSFC Grant No. 62202450, Huawei New IP open identification resolution system project No. TC20201119008 and Postdoctoral Exchange Program No. YJ20210185.

References

1. Alvim, M.S., Chatzikokolakis, K., Palamidessi, C., Pazii, A.: Metric-based local differential privacy for statistical applications. arXiv preprint (2018)
2. Avent, B., Korolova, A., Zeber, D., Hovden, T., Livshits, B.: BLENDER: enabling local search with a hybrid differential privacy model. In: USENIX Security (2017)
3. Bassily, R., Nissim, K., Stemmer, U., Guha Thakurta, A.: Practical locally private heavy hitters. NIPS **30** (2017)
4. Bodon, F.: A fast apriori implementation. In: FIMI. vol. 3, p. 63 (2003)
5. Chatzikokolakis, K., Andrés, M.E., Bordenabe, N.E., Palamidessi, C.: Broadening the scope of differential privacy using metrics. In: PoPETs. pp. 82–102 (2013)
6. Chen, R., Li, H., Qin, A.K., Kasiviswanathan, S.P., Jin, H.: Private spatial data aggregation in the local setting. In: ICDE (2016)
7. Cheu, A., Smith, A., Ullman, J., Zeber, D., Zhilyaev, M.: Distributed differential privacy via shuffling. In: EUROCRYPT. pp. 375–403 (2019)

8. Cormode, G., Maddock, S., Maple, C.: Frequency estimation under local differential privacy. PVLDB **14**(11), 2046–2058 (2021)
9. Cormode, G., Muthukrishnan, S.: An improved data stream summary: the count-min sketch and its applications. Journal of Algorithms **55**(1), 58–75 (2005)
10. Ding, B., Kulkarni, J., Yekhanin, S.: Collecting telemetry data privately. NIPS **30** (2017)
11. Duchi, J.C., Jordan, M.I., Wainwright, M.J.: Local privacy and statistical minimax rates. In: FOCS. pp. 429–438 (2013)
12. Dudek, G.: Aol search log (2007), http://www.cim.mcgill.ca/˜dudek/206/Logs/AOL/
13. Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: TCC. pp. 265–284 (2006)
14. Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., Thakurta, A.: Amplification by shuffling: from local to central differential privacy via anonymity. In: SODA (2019)
15. Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: randomized aggregatable privacy-preserving ordinal response. In: CCS (2014)
16. Gursoy, M.E., Tamersoy, A., Truex, S., Wei, W., Liu, L.: Secure and utility-aware data collection with condensed local differential privacy. TDSC (2019)
17. McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: SIGMOD. pp. 19–30 (2009)
18. Piao, C., Hao, Y., Yan, J., Jiang, X.: Privacy protection in government data sharing: an improved ldp-based approach. SOCA (2021)
19. Qin, Z., Yang, Y., Yu, T., Khalil, I., Xiao, X., Ren, K.: Heavy hitter estimation over set-valued data with local differential privacy. In: CCS. pp. 192–203 (2016)
20. Team, D.P.: Learning with privacy at scale. Apple Machine Learning Journal **1**(8) (2017)
21. Vepakomma, P., Pushpita, S.N., Raskar, R.: DAMS: meta-estimation of private sketch data structures for differentially private COVID-19 contact tracing. Tech. rep., Tech. Rep. (2021)
22. Wang, S., Huang, L., Nie, Y., Wang, P., Xu, H., Yang, W.: Privset: set-valued data analyses with locale differential privacy. In: INFOCOM. pp. 1088–1096 (2018)
23. Wang, S., Qian, Y., Du, J., Yang, W., Huang, L., Xuy, H.: Set-valued data publication with local privacy: Tight error bounds and efficient mechanisms. PVLDB (2020)
24. Wang, T., Blocki, J., Li, N., Jha, S.: Locally differentially private protocols for frequency estimation. In: USENIX Security Symposium. pp. 729–745 (2017)
25. Wang, T., Chen, J.Q., Zhang, Z., Su, D., Cheng, Y., Li, Z., Li, N., Jha, S.: Continuous release of data streams under both centralized and local differential privacy. In: CCS (2021)
26. Wang, T., Li, N., Jha, S.: Locally differentially private frequent itemset mining. In: S&P. pp. 127–143 (2018)
27. Wang, T., Li, N., Jha, S.: Locally differentially private heavy hitter identification. TDSC **18**(2), 982–993 (2019)
28. Warner, S.L.: Randomized response: a survey technique for eliminating evasive answer bias. Journal of the American Statistical Association **60**(309), 63–69 (1965)
29. Yiwen, N., Yang, W., Huang, L., Xie, X., Zhao, Z., Wang, S.: A utility-optimized framework for personalized private histogram estimation. IEEE TKDE (2018)
30. Zhou, M., Wang, T., Chan, T.H., Fanti, G., Shi, E.: Locally differentially private sparse vector aggregation. In: S&P (2022)