Computer Modeling in
Engineering & Sciences

Tech Science Press

DOI: 10.32604/cmes.2023.045400

ARTICLE

KSKV: Key-Strategy for Key-Value Data Collection with Local Differential Privacy

Dan Zhao[1], **Yang You**[2], **Chuanwen Luo**[3,*], **Ting Chen**[4,*] **and Yang Liu**[5]

[1]Artificial Intelligence Development Research Center, Institute of Scientific and Technical Information of China, Beijing, 100038, China

[2]Industry Development Department, NSFOCUS Inc., Beijing, China

[3]School of Information Science and Technology, Beijing Forestry University, Beijing, 100083, China

[4]School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, China

[5]Institute of Computing Technology, China Academy of Railway Sciences Corporation Limited, Beijing, 10081, China

*Corresponding Authors: Chuanwen Luo. Email: chuanwenluo@bjfu.edu.cn; Ting Chen. Email: brokendragon@uestc.edu.cn

Received: 25 August 2023 Accepted: 08 November 2023 Published: 11 March 2024

ABSTRACT

In recent years, the research field of data collection under local differential privacy (LDP) has expanded its focus from elementary data types to include more complex structural data, such as set-value and graph data. However, our comprehensive review of existing literature reveals that there needs to be more studies that engage with key-value data collection. Such studies would simultaneously collect the frequencies of keys and the mean of values associated with each key. Additionally, the allocation of the privacy budget between the frequencies of keys and the means of values for each key does not yield an optimal utility tradeoff. Recognizing the importance of obtaining accurate key frequencies and mean estimations for key-value data collection, this paper presents a novel framework: the Key-Strategy Framework for Key-Value Data Collection under LDP. Initially, the Key-Strategy Unary Encoding (KS-UE) strategy is proposed within non-interactive frameworks for the purpose of privacy budget allocation to achieve precise key frequencies; subsequently, the Key-Strategy Generalized Randomized Response (KS-GRR) strategy is introduced for interactive frameworks to enhance the efficiency of collecting frequent keys through group-and-iteration methods. Both strategies are adapted for scenarios in which users possess either a single or multiple key-value pairs. Theoretically, we demonstrate that the variance of KS-UE is lower than that of existing methods. These claims are substantiated through extensive experimental evaluation on real-world datasets, confirming the effectiveness and efficiency of the KS-UE and KS-GRR strategies.

KEYWORDS

Key-value; local differential privacy; frequency estimation; mean estimation; data perturbation

1 Introduction

Enterprises frequently collect data for analysis with the aim of enhancing the quality of their services. However, such data often contain sensitive information, necessitating robust privacy protection measures. Differential Privacy (DP) [1] has become the *de facto* standard for privacy preservation, ensuring the security of data irrespective of an adversary's background knowledge or computational power. In environments lacking a trusted aggregator, the Local Differential Privacy (LDP) offers robust privacy assurances during the data collection phase. Under LDP, users independently apply a privacy-preserving mechanism to their data before transmission to the aggregator. Notably, several leading enterprises, including Google [2], Apple [3], and Microsoft [4], have implemented LDP.

Initial research in LDP primarily addressed basic statistical data types, focusing on frequency estimation of discrete variables and mean estimation of continuous variables. Further studies extended the paradigm to more sophisticated structured data, including set values [5–7] and graph data [8,9]. Despite this progress, investigations into key-value data queries remain relatively sparse [10–12]. Key-value data, prevalent in practice, necessitates simultaneous analysis of key frequencies and the means of values associated with each key. The accuracy of multidimensional data collection under LDP is particularly challenging due to the constraints of the privacy budget. Moreover, traditional LDP algorithms can significantly distort the intrinsic associations among multidimensional data. The significance of key-value data in big data analytics cannot be overstated, and the correlation between keys and their associated values is typically strong. Thus, the collection of key-value data within the stringent privacy budget constraints of LDP presents a substantial and noteworthy challenge. For example, as illustrated in Table 1, to accurately determine a movie's rating, a sufficient number of reviews, or the frequency of keys, is first required. Subsequently, the mean of these key values represents the movie's rating. Accurate key frequencies not only enable more precise value calculations (since the value under each key is derived from the sum of all values divided by the key's frequency) but also facilitate a more accurate ranking of keys.

Table 1: Example for key-value

Users	Movies and ratings
user 1	<'Forrest Gump', 9>, <'Coco', 8.5>, <'Flipped', 8>
user 2	<'Coco', 9>, <'The Godfather', 9>
user 3	<'Flipped', 8.5>, <'Wonder', 8>, <'Green Book', 9>
...	...

Motivations. Effective management of key-value collections critically relies on two principal factors: the frequency of keys and the mean of the values associated with each key. The primary challenge lies in maximizing the utility of key-value data within the LDP framework, while maintaining robust privacy guarantees. In LDP models, the privacy budget parameter, denoted as ε, is pivotal, serving as a quantitative measure of privacy protection. Given the dual nature of key-value pairs, the privacy budget is judiciously allocated to protect the confidentiality of both the keys and their corresponding values, without compromising their intrinsic correlation. Ye et al. [10] originally proposed a method that equally divided the privacy budget between keys and values; however, this approach did not result in an optimized budget allocation within the LDP paradigm. Later, Gu et al. [12] introduced a unified privacy budget allocation strategy, employing a shared privacy budget for keys and values. Nevertheless, in the context of key-value data, an inaccurate selection of keys can render the associated values meaningless. As such, this paper advocates for a Joint Privacy Budget mechanism, oriented towards enhancing the accuracy of keys. This method not only substantiates the precision of key identification but also improves value computation by: (1) satisfying key frequency estimation requirements, and (2) enabling more accurate mean estimation due to the enhanced accuracy of frequency estimation.

Contribution. This paper aims to address the existing challenges by introducing innovative mechanisms that prioritize key strategies, aimed at increasing the utility of key-value data collection in both interactive and non-interactive environments. For non-interactive frameworks, we propose the Key-Strategy Unary Encoding (KS-UE), strategically allocating a predominant portion of the privacy budget to key frequencies to enhance utility. In interactive environments, we employ the Key-Strategy Generalized Randomized Response (KS-GRR), which uses a group-and-iteration approach across various tasks to yield superior estimation results. Furthermore, the paper thoroughly examines the flexibility of these strategies in situations where users hold either single or multiple key-value pairs.

The salient contributions of this work can be summarized as follows:

- Acknowledging the paramountcy of accurate keys in key-value collection, we advance the key-strategy mechanisms KS-UE for non-interactive frameworks and KS-GRR for interactive frameworks within the confines of LDP.
- By leveraging KS-UE and KS-GRR, we formulate algorithms tailored for contexts wherein users hold either singular or multiple key-value pairs, enhancing the adaptability of our approach.
- We provide rigorous theoretical substantiation for the minimal variance of KS-UE. Additionally, empirical evaluations corroborate that strategies KS-UE and KS-GRR can ascertain more precise frequent keys while maintaining the integrity of mean accuracy.

Roadmap. Section 2 will elaborate on related works. Subsequently, Section 3 will provide background information and the theoretical foundation for our study. In Section 4, we introduce our key-strategy algorithms for single and multiple key-value pairs within the framework of Local Differential Privacy. Section 5 will present the experimental results, while Section 6 will conclude the paper.

2 Related Work

Local Differential Privacy is a potent technique employed to safeguard sensitive user data during the data collection process. The fundamental mechanism of LDP yields a probability distribution as output rather than revealing authentic statistical information. Randomized Response (RR), first introduced in the 1960s, is recognized as a precursor to LDP methods [13] and was later formalized as a local privacy model by Dwork [14,15]. Subsequent research has led to the development of methodologies such as RAPPOR [2], Optimal-RR (O-RR), Optimal-RAPPOR (O-RAPPOR) [16], Sampled Histogram (SH) [17], Optimal Unary Encoding (OUE), and Optimal Local Hashing (OLH) [18] for singleton frequency estimation. Additional studies [4,19] have focused on mean estimation with numerical attributes. More complex data structures such as set values [5,6], marginal release [20,21], numerical values [19,22], graph data [8,9,23,24], spatiotemporal data [1,25], time-series [26], distribution estimation [27], range queries [28], and machine learning [29,30] have been examined. Further, strategies to counteract attacks have been proposed [31–33]. Nevertheless, only a limited number of studies [10–12,34] have concentrated on key-value data queries.

Ye et al. [10] were the first to formalize two tasks in key-value data collection under LDP: estimating the frequencies of keys and the mean of values under each key. They proposed PrivKV for non-interactive frameworks and later introduced optimized versions, PrivKVM and PrivKVM+, for interactive frameworks. Ye et al. further improved PrivKVM+ in [34]. Sun et al. [11] offered another approach within the PrivKV framework, but they did not analyze the impact of the correlation between perturbations on the tighter budget composition. Gu et al. [12] addressed this issue by proposing an optimized budget allocation and an advanced 'padding-and-sampling' mechanism. They used budget allocation to optimize privacy parameters for better estimation results. In this paper, we propose a more effective scheme, termed key-strategy frameworks, for key-value data collection under LDP.

3 Preliminaries

In this section, we initially delineate the problem, followed by an introduction to the concept of LDP. The notations used in this paper are summarized in Table 2.

Table 2: Notations

Notation	Definition
ε	Privacy budget
$\mathscr{K}$	The domain of keys
u^i	The i-th user
P^i	The key-value pair of user i
S^i	The key-value pair set of user i
$\mathbf{b}^i$	The original unary encoding vector of user i
$\mathbf{b}^{i*}$	The perturbed unary encoding vector of user i
f_k	The relative frequency of key k
m_k	The mean of values under key k
TKS	Top frequent keys set

3.1 Problem Definition and Challenge

Key-Value Data Collection. The issue of key-value data collection under LDP was first put forth by [10]. Assume there are n users (denoted as $u^i \in \mathscr{U}$ where $i = 1, 2, ..., n$). Each user u^i possesses either a single or a set of key-value data. The key-value pair is denoted as $P^i :=< k^i, v^i >$ (where $k \in \mathbb{K}$ and $v \in \mathbb{V}$), forming a domain $\mathbb{P}$. The domain size of the key is $|K| = d$ and the domain of the value v is within the range of $[-1, 1]$. The objective of the aggregator is to estimate the statistical data of the users, specifically, the frequency of keys and the mean of values under each key.

- **Frequency Estimation:** The purpose of this task is to estimate the frequency of each key k, defined as: $\hat{f}_k = \dfrac{|\{P^i|\exists k \in P^i\}}{n}$
- **Mean Estimation:** The goal of this task is to estimate the mean of values associated with each key k, defined as: $\hat{m}_k = \dfrac{\sum_{i=1,k\in P^i}^{n} v^i}{n \cdot \hat{f}_k}$

In practical application scenarios, most users possess multiple values $P^i := \{< k^{i1}, v^{i1} >, < k^{i12}, v^{i2} >, ..., < k^{il}, v^{il} >\}$. The existence of multiple key-value pairs implies an added dimension to key-value data, symbolizing a synthesis of set-value and key-value data. The aggregator also estiamte the statistical data $\hat{f}_k$ and $\hat{m}_k$. This amalgamation is also one of the potential developmental trajectories in LDP.

3.2 Local Differential Privacy

LDP is a localized model of DP intended for data collection without a trustworthy aggregator [5,18,35,36]. An LDP algorithm, denoted as Ψ, ensures that the probability of one value being transmitted to the aggregator closely approximates the probability of any other values being sent. The notion of ε-LDP is defined as follows.

Definition 3.1 (ε-local differential privacy.)**.** A randomized mechanism Ψ with domain $Dom(\Psi)$ and range $Ran(\Psi)$ is ε-local differential private if for any input items $v, v' \in Dom(\Psi)$, and any output $s \in Ran(\Psi)$, the following inequality holds:

$$\frac{Pr[\Psi(v) = s]}{Pr[\Psi(v') = s]} \le e^{\varepsilon} \tag{1}$$

The privacy budget, ε, can be adjusted to strike a balance between data availability and privacy intensity. LDP can offer a stronger level of privacy protection than a centralized framework as each user reports only the perturbed data.

3.3 Mechanism under LDP

We assume the size of domain $\mathscr{D}$ is d, each user has one item of $\mathscr{D}$.

Randomized Response(RR)/Generalized Randomized Response(GRR). The perturbation function of GRR [18] is

$$Pr[\Psi(v) = s] = \begin{cases} \dfrac{e^{\varepsilon}}{e^{\varepsilon} + d - 1}, & if \quad s = v \\ \dfrac{1}{e^{\varepsilon} + d - 1}, & if \quad s \neq v \end{cases}$$

where RR is the special situation $d = 2$.

Unary Encoding (UE). [18] Each user transforms her item into a vector of length d, where only the mapped position is 1, and all others are 0s. Subsequently, each bit is independently perturbed through the function

$$Pr[\Psi(v^*[j]) = 1] = \begin{cases} p, & if \quad v[j] = 1 \\ q, & if \quad v[j] = 0 \end{cases}$$

where $p + q$ need not be equal to 1.

When $p = \dfrac{1}{2}$ and $q = \dfrac{1}{e^{\varepsilon} + 1}$, the variance of frequency is minimized; therefore, under these conditions, it is termed as Optimized Unary Encoding (OUE).

Optimal Local Hashing (OLH). [18] The OLH protocol was developed to ameliorate the challenges posed by attributes with extensive categories. Initially, the client-side algorithm transforms the user's actual value v to a diminished hash value domain g utilizing a precise hash function. Subsequent to this transformation, the algorithm applies a randomized response to the hash value within this condensed domain. The parameter g represents a balanced compromise between the information loss occurring during the hashing and the subsequent randomization step; the balance is deemed optimal when $g = e^{\varepsilon} + 1$. The temporal complexity of this algorithm is characterized as $O(\log n)$, while the spatial complexity is quantified as $O(n \log |D|)$.

3.4 Padding-and-Sampling Protocol

Originally utilized for itemset data [6], the 'padding-and-sampling' protocol was first adopted for key-value data in [12]. In this setup, each user samples a single key-value pair from their set. A crucial parameter, denoted as l, ensures a consistent sampling rate while l dummy key-value pairs are generated. A key-value pair from the user's set Si is selected randomly with a probability of $\frac{|S^i|}{max\{|S^i|, l\}}$, and a dummy key-value pair is chosen with the remaining probability.

3.5 Competitor

PrivKV. Ye et al. [10] proposed PrivKV and improved version PrivKVM and PrivKVM+ on key-value data collection under LDP first time. The primary process comprises: (a) encoding the key-value pair to a position of a vector, leaving all other positions as $< 0, 0 >$; (b) the key is perturbed first at each vector position, followed by the value perturbation, with the privacy budget split equally between them.

PCKV-UE and PCKV-GRR. Gu et al. [12] proposed PCKV-UE and PCKV-GRR, leveraging the privacy budget in a more efficient manner. They employed a joint perturbation strategy, rather than dividing the privacy budget evenly. The value's perturbation is contingent upon the key's perturbation, which optimizes the privacy budget composition. Both methods, theoretically and experimentally, outperform those proposed by Ye et al. Thus, we have excluded the PrivKV series from our experiment. The variances of keys and means in PCKV-UE are as follows:

$$Var(\hat{f_k}) = \frac{8(e^{\varepsilon}+1)}{(e^{\varepsilon}-1)^2 n} + \frac{f_k}{n}$$

$$Var[\hat{m}_k] \leq \frac{8(e^{\varepsilon}+1)^2 n}{(e^{\varepsilon}+3)(e^{\varepsilon}-1)^2 n_k^2} + \frac{2(e^{\varepsilon}+1)^2}{(e^{\varepsilon}+3)(e^{\varepsilon}-1)n_k}$$

It is noteworthy that both theoretically and experimentally, Gu et al. have demonstrated that PCKV-UE and PCKV-GRR are superior to the series of PrivKV algorithms under equivalent conditions. Even after the conditions in [34] were collectively altered, PCKV-UE and PCKV-GRR still showcased excellent performance. Therefore, this paper solely compares PCKV-UE and PCKV-GRR.

For multiple key-value. In [12], when PCKV-UE and PCKV-GRR address multiple key-value pairs, they pre-set an average length l, yielding favorable results for datasets conforming to this specified length, while performance deteriorates for those that do not align with the set length. In contrast, PrivKV [10], when dealing with multiple key-value pairs, directly employs a "padding-and-sampling" approach to simplify multidimensional data into unidimensional data. Given that PCKV-UE and PCKV-GRR outperform PrivKV significantly, this paper solely compares the superior PCKV-UE and PCKV-GRR algorithms.

4 Key-Strategy and Mechanisms

Our mechanisms aim to gather the frequencies of keys and the mean of values under each key. If most keys have low frequencies, the mean values corresponding to these keys are largely irrelevant. Hence, if the domain is large, it becomes essential to reduce the domain in order to find top frequent keys. In this section, we introduce two key-strategy mechanisms for a **singleton** key-value pair under LDP: key-strategy unary encoding (KS-UE) in non-interactive frameworks and key-strategy generalized randomized response (KS-GRR) in interactive frameworks. When each user possesses

multiple key-value pairs, padding-sampling is utilized to convert them into a singleton key-value pair. Subsequently, each user can implement the aforementioned two mechanisms.

4.1 KS-UE for Singleton Pair in Non-Interactive Frameworks

In non-interactive frameworks, users transmit their perturbed data to the aggregator using a one-way communication model. Here, the unary encoding (UE) mechanism comes into play, encoding each pair into a bit vector with only the bit corresponding to the key set to a value, and all others to zero. Building upon previous algorithms, KS-UE skews the privacy budget in favor of key frequencies to improve utility. The KS-UE methodology unfolds in three distinct phases: *Discretization*, *Encoding and Perturbation*, and *Decoding*, as delineated in Algorithm 2.

Discretization. To alleviate communication and time complexity, the initial step discretizes the value of the key-value pair from float type to binary $\{1, -1\}$ using formula (2). Consequently, the domain for all key-value pairs becomes $\{<1, 1>, <1, -1>, <0, 0>\}$. In an attempt to further reduce communication costs, we simplify them to three single values $\{1, -1, 0\}$. If a value is zero, the key is also set to zero; otherwise, the key is set to one.

$$v^* = \begin{cases} 1 & w.p. \quad \dfrac{1+v}{2} \\ -1 & w.p. \quad \dfrac{1-v}{2} \end{cases} \tag{2}$$

Encoding and Perturbation. The first operation here is to employ UE to create a bit vector $\mathbf{b}$ of size d. The position in $\mathbf{b}$ corresponding to the key is v^*, and all other positions are set to zero. Subsequently, each bit of the vector undergoes perturbation in accordance with formula (3) ($\mathbf{b}[j] = 0$) and formula (4) ($\mathbf{b}[j] \neq 0$).

```
Algorithm 1: Key-Strategy Unary Encoding (KS-UE)
Require: The privacy budget $\varepsilon$; the set of key $\mathscr{K}$ ($|\mathscr{K}| = d$); the perturbed parameters $a,p$ from privacy budget $\varepsilon$.
Ensure: The estimated frequency $f_k$ of key $k$ and the mean $m_k$ under this key;
User Side:
for each user $u^i$ with value $< k^i, v^i >$, $i = 1$ to $n$ do
    Discretize $v^{i^*} \leftarrow v^i$ (see formula (2));
    Generate vector $\mathbf{b}^i$, where the $\mathbf{b}^i[k_{index}] = v^{i^*}$, and others are 0s;
    Perturb $\mathbf{b}^{i^*} \leftarrow \mathbf{b}^i$, where the perturbation of each bit follows formulas (3) and (4);
    Send $\mathbf{b}^{i^*}$ to the aggregator;
end for
Aggregator Side:
  Collects three statistic results: $n_k^{-1}$, $n_k^1$ and $sum^k$;
  Calculate the estimation: $\hat{f}_k = \dfrac{(n_k^{-1} + n_k^1)/n - a}{1 - p - a}$ and $\hat{m}_k = \dfrac{n_k^1 - n_k^{-1}}{(3p-1)\hat{f}_k n}$.
  return $\hat{f}_k$ and $\hat{m}_k$;
```

$$\mathbf{b}^*[j] = \begin{cases} 1 & w.p. \quad 0.5a \\ -1 & w.p. \quad 0.5a \\ 0 & w.p. \quad 1-a \end{cases} \tag{3}$$

$$\mathbf{b}^*[j] = \begin{cases} \mathbf{b}[j] & w.p. \quad p \\ -\mathbf{b}[j] & w.p. \quad 1-2p \\ 0 & w.p. \quad p \end{cases}, \tag{4}$$

where $p = \dfrac{e^{\varepsilon}+1}{2(e^{\varepsilon}+2)}$ and $a = \dfrac{2}{e^{\varepsilon}+2}$ $\left(\dfrac{1}{3} < p < \dfrac{1}{2}\text{and}0 < a < \dfrac{1}{2}\right)$. Finally, $\mathbf{b}^*$ is sent to the aggregator. According to the above perturbation strategy, the disturbed vector $\mathbf{b}^*$ can be obtained in client-side and sent to the aggregator.

Decoding. The aggregator collects $\mathbf{b}^*$ from each user. Then, the aggregator need to calculate three statistic results n_k^{-1}, n_k^1 and sum_k. The aggregator collects $\mathbf{b}^*$ from each user. Then, the aggregator computes three statistical results: n_k^{-1}, n_k^1 and sum_k.

$$\begin{cases} n_k^{-1} := Count(\mathbf{b}[k_{index}] = -1) \\ n_k^1 := Count(\mathbf{b}[k_{index}] = 1) \\ sum_k := n_k^1 - n_k^{-1} \end{cases},$$

where n_k^{-1} and n_k^1 represent the count of 1 and -1 under key k respectively, and sum_k indicates the summation of values under key k. Finally, the aggregator can estimate $\hat{f}_k$ and $\hat{m}_k = sum_k/\hat{f}_k$.

$$\hat{f}_k = \frac{(n_k^{-1} + n_k^1)/n - a}{1 - p - a} \tag{5}$$

Having estimated the frequency of key k, the mean value under key k, $\hat{m}_k$ can be estimated.

$$\hat{m}_k = \frac{n_k^1 - n_k^{-1}}{(3p-1)\hat{f}_k n}. \tag{6}$$

4.2 KS-GRR for Singleton Pair in Interactive Frameworks

Within interactive frameworks, users are systematically assembled into random group, and the entirety of the privacy budget is meticulously allocated to facilitate involvement in two distinct tasks for each assembly, as depicted in Fig. 1: (1) the estimation of the most prevalent keys within key-value pairs; (2) the computation of the frequencies of the aforementioned prevalent keys, as well as the mean of the values corresponding to these keys. Then, the aggregator is enabled to process and extrapolate the preliminary results from the initial group, subsequently deriving the conclusive frequencies of the superior keys and the corresponding mean values from the subsequent group. The KS-GRR model employs a sophisticated grouping methodology, aiming to optimize the efficacy of the data collection process, thereby contributing to enhanced accuracy and utility in the realization of key-value collection goals.

Top frequent keys. In the context of top frequent keys, the value in key-value pairs is deemed irrelevant, as the focus here is solely on estimating the frequencies of keys. Thus, this stage transforms the collection of key-value pairs into a singleton keys collection under LDP, directly utilizing the OLH algorithm. To obtain the top-t frequent keys, the top frequent-$2t$ keys should be identified and used as a candidate set TKS in this task.

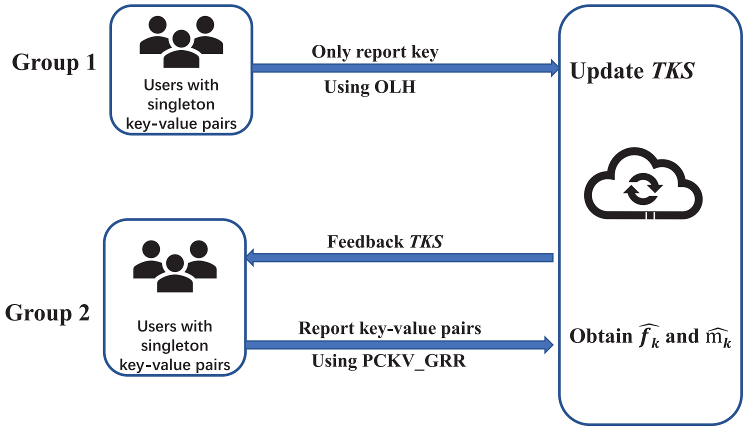

Group 1
Users with singleton key-value pairs
Only report key
Using OLH
Update *TKS*
Feedback *TKS*
Group 2
Users with singleton key-value pairs
Report key-value pairs
Using PCKV_GRR
Obtain $\widehat{f_k}$ and $\widehat{m}_k$

Figure 1: Illustration of KS-GRR for singleton key-value pair

Frequency and mean. The aggregator transmits the top-$2t$ candidate keys set TKS to the remaining users for final estimation. If the key is within TKS, each user retains their key-value pair; otherwise, a dummy key-value pair $<\bot, \pm 1>$ is used to replace the original key-value pair. Since the potential number of keys is currently $2t+1$, we set $TKS \cap \{\bot\}$ as $\mathscr{K}'$. Subsequently, discretizes the value of the key-value pair from float type to binary $\{1, -1\}$ using formula (2). Then, each user with value $<k^i, v^i>$ is perturbed using formula (7) and sent to the aggregator. The aggregator can estimate $\hat{f}$ of the top-t frequent keys and $\hat{m}$ under these keys through computation.

$$<k^{i*}, v^{i*}> = \begin{cases} <k^i, v^i> & w.p. \quad \dfrac{e^{\varepsilon}}{e^{\varepsilon}+4t+1} \\ <k^i, -v^i> & w.p. \quad \dfrac{1}{e^{\varepsilon}+4t+1} \\ <\alpha, 1> (\alpha \in \mathscr{K}' \backslash k^i) & w.p. \quad \dfrac{1}{e^{\varepsilon}+4t+1} \\ <\alpha, -1> (\alpha \in \mathscr{K}' \backslash k^i) & w.p. \quad \dfrac{1}{e^{\varepsilon}+4t+1} \end{cases} \tag{7}$$

Algorithm 2: Key-Strategy Generalized Randomized Response (KS-GRR)

Require: The privacy budget ε; the set of key $\mathscr{K}$ ($|\mathscr{K}| = d$); the perturbed parameters *a,p* from privacy budget ε.

Ensure: The estimated frequency f_k of key k and the mean m_k under this key;

```
Group users 1:
each user $u^i$ only send $k^i$ to the aggregator through tradition OLH algorithm;
Aggregator Side:
Obtain top-$2t$ candidate frequent keys $TKS$;
Send $TKS$ to Group users 2;
Group users 2:
for each user $u^i$ with value $<k^i, v^{i'}>$, $i = 1$ to $n$ do
    if $k^i \in TKS$ then
```

(Continued)

Algorithm 2 (continued)

```
            Discretize $v^i \leftarrow v^{i'}$ through formula (2);
        else
            Replace $< k^i, v^i >$ to dummy key-value pair $<\perp, \pm 1>$;
        end if
        Perturb $< k^{i*}, v^{i*} > \leftarrow < k^i, v^i >$ through formula (7);
        Send $< k^{i*}, v^{i*} >$ to the aggregator;
    end for
    Aggregator Side:
    Collects three statistic results: $n_k^{-1}$, $n_k^1$ and $sum^k$;
    Calculate the estimation: $\hat{f}_k = \frac{(n_k^{-1} + n_k^1)/n - 2q}{p - q}$ and $\hat{m}_k = \frac{n_k^1 - n_k^{-1}}{(p + q)\hat{f}_k n}$, where
        $p = \frac{e^\varepsilon}{e^\varepsilon + 4t + 1}, q = \frac{1}{e^\varepsilon + 4t + 1}$.
    return $\hat{f}_k$ and $\hat{m}_k$;
```

4.3 KS-UE and KS-GRR for Multiple Key-Value Pairs

The scenario of multiple key-value pairs indicates that each user possesses a set of such pairs. In order to accommodate this situation, we employ an advanced protocol known as ‘padding-and-sampling’ as described in Gu et al. [12], specifically designed to enhance the performance of key-value data analysis.

The ‘padding-and-sampling’ protocol necessitates a global parameter l to ensure uniform sampling rate across all users. Consequently, l dummy key-value pairs are introduced into the key-value pairs domain. When $|S^i| < l$, the user u^i incorporates dummy key-value pairs DS to achieve a total of $|S^i \cup DS| = l$. However, the parameter l within the ‘padding-and-sampling’ protocol can lead to biased estimations. The bias escalates with an increase in l, conversely the variance diminishes. Then, user u^i randomly selects one key-value pair to participate in the KS-UE or KS-GRR algorithms.

KS-UE for multiple key-value pairs. In order to allow for a comparison with PCKV-UE, KS-UE employs the same ‘padding-and-sampling’ protocol as PCKV-UE. Consequently, following the ‘padding-and-sampling’ protocol, each user will have a singleton key-value pair. The KS-UE algorithm (discussed in Section 4.1) is utilized to ascertain the frequencies of the top-t frequent keys and the mean under these keys.

KS-GRR for multiple key-value pairs. The process of KS-GRR is recalibrated for the case of multiple key-value pairs, as depicted in Fig. 2. Given the tripartite tasks involved in KS-GRR, the users are accordingly segmented into three groups, each group catering to a specific task.

- *Task 1:* Users in this group bypass the ‘padding-and-sampling’ protocol and opt to randomly select one of their keys to be transmitted to the aggregator via the OLH algorithm. This technique does not ascertain the frequencies of the keys; however, it reliably preserves the ordinal information of the keys. Notably, the estimations remain stable and unaffected by variations in the value of l.
- *Task 2:* For the users engaged in this task, the aggregator provides feedback on the top keys set TKS. This task requires a limited number of users to identify an appropriate l. Moreover, the data essential for this task pertains to the size of each user’s set, not the key-value pair.

Initially, every user generates her new key-value pairs set, ensuring that the keys belong to *TKS*. Subsequently, each user submits the size of the private set, and the aggregator identifies the smallest l such that $\frac{\sum_{ls=1}^{l} \Phi_{OLH}(ls)}{\sum_{ls=1}^{all} \Phi_{OLH}(ls)} > 0.9$.

- *Task 3:* For the users participating in this task, the aggregator provides feedback on the top keys set *TKS* as well as the adjusted l. Users maintain key-value pairs if the keys are found in *TKS*, and subsequently implement the 'padding-and-sampling' protocol with the l determined in task 2. This facilitates the estimation of the frequencies of top frequent keys and the mean under these keys.

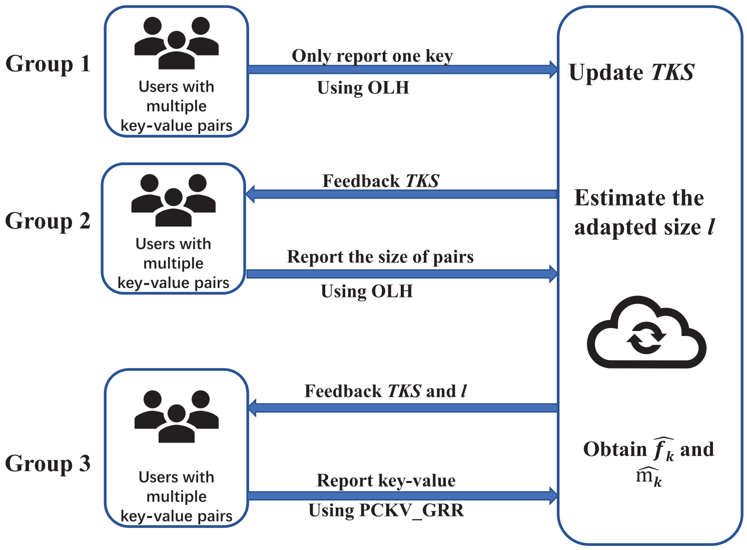

Group 1
Users with multiple key-value pairs
Only report one key
Using OLH
Update *TKS*
Group 2
Users with multiple key-value pairs
Feedback *TKS*
Estimate the adapted size *l*
Report the size of pairs
Using OLH
Group 3
Users with multiple key-value pairs
Feedback *TKS* and *l*
Obtain $\widehat{f_k}$ and $\widehat{m_k}$
Report key-value
Using PCKV_GRR

Figure 2: Illustration of KS-GRR for multiple key-value pairs

4.4 Privacy and Analysis

In this section, we will initially provide evidence to confirm that KS-UE satisfies ε-LDP. Subsequently, we will conduct an analysis on the estimation error of KS-UE. The error yielded by KS-UE is demonstrably less than that produced by preceding algorithms.

Theorem 4.1. KS-UE satisfies ε-LDP.

Proof. Let $\mathscr{S}$ be a key-value pair set, and each key-value pair $< k, v > \in \mathscr{S}$ has domain $k \in \mathscr{K}$ and $v \in \{1, -1\}$. For any two key-value pairs $P^i = < k^i, v^i >$ and $P^j = < k^j, v^j >$ $(P^i, P^j \in \mathscr{S})$ has the maximum different probability of outputting a same vector when $k^i \neq k^j$.

$$Pr_{upper} = Pr\left(\mathbf{b}^*[i] = v^i, \mathbf{b}^*[j] = 0 \middle| \mathbf{b}[i] = v^i, \mathbf{b}[j] = 0\right) = p(1-a)$$

$$Pr_{lower} = Pr\left(\mathbf{b}^*[i] = v^i, \mathbf{b}^*[j] = 0 \middle| \mathbf{b}[i] = 0, \mathbf{b}[j] = \pm v^i\right) = \frac{1}{2}ap$$

$$= Pr\left(\mathbf{b}^*[i] = v^i, \mathbf{b}^*[j] = 0 \middle| \mathbf{b}[i] = -v^i, \mathbf{b}[j] = \pm v^i\right) = (1-2p)p$$

Then, P^i and P^j are indistinguishable at most through KS-UE:

$$\frac{Pr(\mathbf{y}|P^i)}{Pr(\mathbf{y}|P^j)} \le \frac{Pr_{upper}}{Pr_{lower}} = e^{\varepsilon}$$

Theorem 4.2. (Estimation Error Analysis). The expectation and variances of $\hat{f}_k$ and $\hat{m}_k$ are as following:

$$\mathbb{E}(\hat{f}_k) = f_k \tag{8}$$

$$Var(\hat{f}_k) = \frac{8e^{\varepsilon}}{(e^{\varepsilon}-1)^2 n} + \frac{(e^{\varepsilon}-3)f_k}{(e^{\varepsilon}-1)n} \tag{9}$$

$$\mathbb{E}[\hat{m}_k] \approx \frac{\sum^{n} v_k^i}{n_k}\left[1 + \frac{8e^{\varepsilon}n}{(e^{\varepsilon}-1)^2 n_k^2} - \frac{4}{(e^{\varepsilon}-1)n_k}\right] \tag{10}$$

$$Var[\hat{m}_k] \le \frac{8(e^{\varepsilon}+2)n}{(e^{\varepsilon}-1)^2 n_k^2} + \frac{2(e^{\varepsilon}+2)}{(e^{\varepsilon}-1)n_k} \tag{11}$$

Proof. For formula (8). For convenience, let n_k denotes the real count of key k and $n_k^* = n_k^1 + n_k^{-1}$, then $\hat{n}_k = \frac{n_k^* - na}{1-a-p}$. We first prove the unbiasedness.

$$\mathbb{E}(\hat{n}_k) = \frac{\mathbb{E}(n_k^1) + \mathbb{E}(n_k^1) - na}{1-a-p} = \frac{(pn_k + (n-n_k)\frac{1}{2}a) + ((1-2p)n_k + (n-n_k)\frac{1}{2}a) - na}{1-a-p}$$

$$= \frac{n_k(p - \frac{1}{2}a + 1 - 2p - \frac{1}{2}a)}{1-a-p} = n_k$$

Therefore, the relative frequency $\mathbb{E}(\hat{f}_k) = \mathbb{E}\frac{\hat{m}_k}{n} = f_k$.

For formula (9). Next, we calculate the variance of $\hat{f}_k$:

$$Var(\hat{n}_k) = \sum^{n_k}(1-p)p + \sum^{n-n_k}(1-a)a = n_k(p-p^2) + (n-n_k)(a-a^2)$$

$$= n\frac{2e^{\varepsilon}}{(e^{\varepsilon}+2)^2} + n_k\left(\frac{(e^{\varepsilon}-1)(e^{\varepsilon}-3)}{4(e^{\varepsilon}+2)^2}\right)$$

$$\therefore Var(\hat{f}_k) = \frac{Var(\hat{n}_k)}{n^2(1-a-p)^2} = \frac{n_k(p-p^2) + (n-n_k)(a-a^2)}{n^2(1-a-p)^2}$$

$$= \frac{4(e^{\varepsilon}+2)^2}{n^2(e^{\varepsilon}-1)^2}\left[n\frac{2e^{\varepsilon}}{(e^{\varepsilon}+2)^2} + n_k\left(\frac{(e^{\varepsilon}-1)(e^{\varepsilon}-3)}{4(e^{\varepsilon}+2)^2}\right)\right] = \frac{8e^{\varepsilon}}{(e^{\varepsilon}-1)^2 n} + \frac{(e^{\varepsilon}-3)f_k}{(e^{\varepsilon}-1)n}$$

For formulas (10) and (11). We calculate the expectation and variance of mean estimation. From the multivariate Taylor Expansions of functions of random variables, the expectation of quotient of two random variables $\mathbf{X}$ and $\mathbf{Y}$ can be approximated by

$$\begin{aligned} &\mathbb{E}\left[\frac{X}{Y}\right] \approx \frac{\mathbb{E}[X]}{\mathbb{E}[Y]} - \frac{Cov_{X,Y}}{\mathbb{E}[Y]^2} + \frac{\mathbb{E}[X]}{\mathbb{E}[Y]^3} \cdot Var[Y] \\ &Var\left[\frac{X}{Y}\right] \approx \frac{Var[X]}{\mathbb{E}[Y]^2} - \frac{2\mathbb{E}[X]Cov_{X,Y}}{\mathbb{E}[Y]^3} + \frac{\mathbb{E}[X]^2}{\mathbb{E}[Y]^4}Var[Y] \end{aligned} \tag{12}$$

For convenience, we denote $X_k = n_k^1 - n_k^{-1}$, $Y_k = n_k^1 + n_k^{-1} + na$, then

$$\hat{m}_k = \frac{X}{Y} \cdot \frac{1-a-p}{3p-1},$$

$$\mathbb{E}[X_k] = (3p-1)\sum^n v^i,$$

$$\mathbb{E}[Y_k] = (1-p-a)n_k$$

The variances are

$$Var[X_k] = na + n_k(1-p-a) - (3p-1)^2\sum^n v^{i^2} = \frac{2}{(e^\varepsilon+2)}n + \frac{e^\varepsilon - 1}{2(e^\varepsilon+2)}n_k - \frac{(e^\varepsilon-1)^2}{4(e^\varepsilon+2)^2}\sum^n v^{i^2}$$

$$Var[Y_k] = n_k[(1-p)p - (1-a)a] + na(1-a) = \frac{(e^\varepsilon-1)(e^\varepsilon-3)}{4(e^\varepsilon+2)^2}n_k + \frac{2e^\varepsilon}{(e^\varepsilon+2)^2}n$$

$$\begin{aligned} Cov_{X,Y} &= Cov[n_k^1 - n_k^{-1}, n_k^1 + n_k^{-1}] = \mathbb{E}[(n_k^1 - n_k^{-1})(n_k^1 + n_k^{-1})] - \mathbb{E}[n_k^1 - n_k^{-1}] \cdot \mathbb{E}[n_k^1 + n_k^{-1}] \\ &= \mathbb{E}[n_k^1 - n_k^{-1}] - [\mathbb{E}[n_k^1]^2 - \mathbb{E}[n_k^{-1}]^2] = Var[n_k^1] - Var[n_k^{-1}] = p \cdot \mathbb{E}(X) \end{aligned}$$

$\mathbb{E}[X]$, $\mathbb{E}[Y]$, $Var[Y]$ and $Cov_{X,Y}$ are computed by their exact values.

$$\begin{aligned} \therefore \mathbb{E}[\hat{m}_k] &= \frac{(1-p-a)\mathbb{E}[\frac{X}{Y}]}{(3p-1)} \\ &\approx \frac{(1-p-a)}{(3p-1)}\left[1 - \frac{p}{\mathbb{E}[Y]} + \frac{Var[Y]}{\mathbb{E}[y]^2}\right] \cdot \frac{\mathbb{E}[X]}{\mathbb{E}[Y]} \\ &= \frac{\sum^n v_k^i}{n_k}\left[1 + \frac{8e^\varepsilon n}{(e^\varepsilon-1)^2 n_k^2} - \frac{4}{(e^\varepsilon-1)n_k}\right] \end{aligned}$$

$$\therefore Var[\hat{m}_k] = \frac{8(e^\varepsilon+2)n}{(e^\varepsilon-1)^2 n_k^2} + \frac{2(e^\varepsilon+2)}{(e^\varepsilon-1)n_k} + \left[\frac{8e^\varepsilon n}{(e^\varepsilon-1)^2 n_k^4} - \frac{(e^\varepsilon+5)}{(e^\varepsilon-1)n_k^3} - \frac{1}{n_k^2}\right]\left(\sum^n v^{i^2}\right)$$

The proof is completed.

5 Experimental Evaluation

In this section, we set up experiments on real datasets to validate our analysis with different approaches.

5.1 Experimental Setup

Datasets. We ran experiments on the following datasets:

- E-Commerce (Ecommerce rating dataset)[1]: This dataset contains the merchant transactions of 23,486 users with 1,206 keys. Each user has only one key-value pair. This dataset is used in both singleton and multiple scenarios.
- Clothing (Clothing fit and rating dataset)[2]: This dataset contains the merchant transactions of 47,959 users with 5,850 keys. Each user has multiple key-value pair. For singleton key-value pair collection, we treat each key-value pair as a individual record. Thus, clothing-singleton dataset has 82,789 records with 5,850 keys.

Metrics. This study aimed to find the frequent itemsets together with their frequencies, which requires different metrics to evaluate their utilities. We adopted the normalized cumulative rank (NCR) [6,18] and squared error (SE) to assess the frequent itemsets and frequencies, respectively.

1). Normalized Cumulative Rank (NCR). The quality function with the most t keys ranked is as follows: $Score(k_j) = t - j + 1$ for j-th highest-ranked key, and all other non-top keys have scores 0. To normalize this into a value between 0 and 1, we divide the sum of scores by the maximum possible score $\left(\text{i.e., } \frac{t(t+1)}{2}\right)$. Thus, a higher score indicates better identification.

2). Squared Error (Var). We measured the estimation accuracy for both frequency and mean by using averaged Mean Squared Error. That is

$$MSE_{freq} = \frac{1}{|\mathscr{X}|}\sum_{k\in\mathscr{X}}(\hat{f}_k - f_k)^2, \qquad MSE_{mean} = \frac{1}{|\mathscr{X}|}\sum_{k\in\mathscr{X}}(\hat{m}_k - m_k)^2$$

where f_k and m_k are the real set of frequency and mean of key k. Note that we account only for top-t frequent keys that are successfully identified by the algorithm, i.e., $\mathscr{X}$ is the intersection of the real top-t keys set and the estimated top-t keys set. Thus, lower variance means more accurate estimation.

Parameter settings. In KS-GRR for multiple pairs, we use 40% of all users to find top-2t frequent keys *TFK*, 10% to report the size of their key-value pairs whose keys are in *TFK*, and 50% to obtain final results.

Selected approaches. We compare our KS-UE and KS-GRR with the following approaches. The approaches used in the evaluation are as follows:

(i) *PCKV-UE* [12]. PVKV-UE is the state-of-the-art approach for key-value pairs collection.

(ii) *PVKV-GRR* [12]. PCKV-GRR can obtain benifit from ‘padding-and-sampling’ than PCKV-UE, but only performs well when $|\mathscr{K}|$ is small.

(iii) *KS-Adap*. KS-Adap is a manifestation for multiple key-value pairs of KS-GRR. We use 40% of all users to find top-2t frequent keys *TFK*, 10% to report the size of their key-value pairs whose keys are in *TFK*, and 50% to obtain final results for multiple pairs.

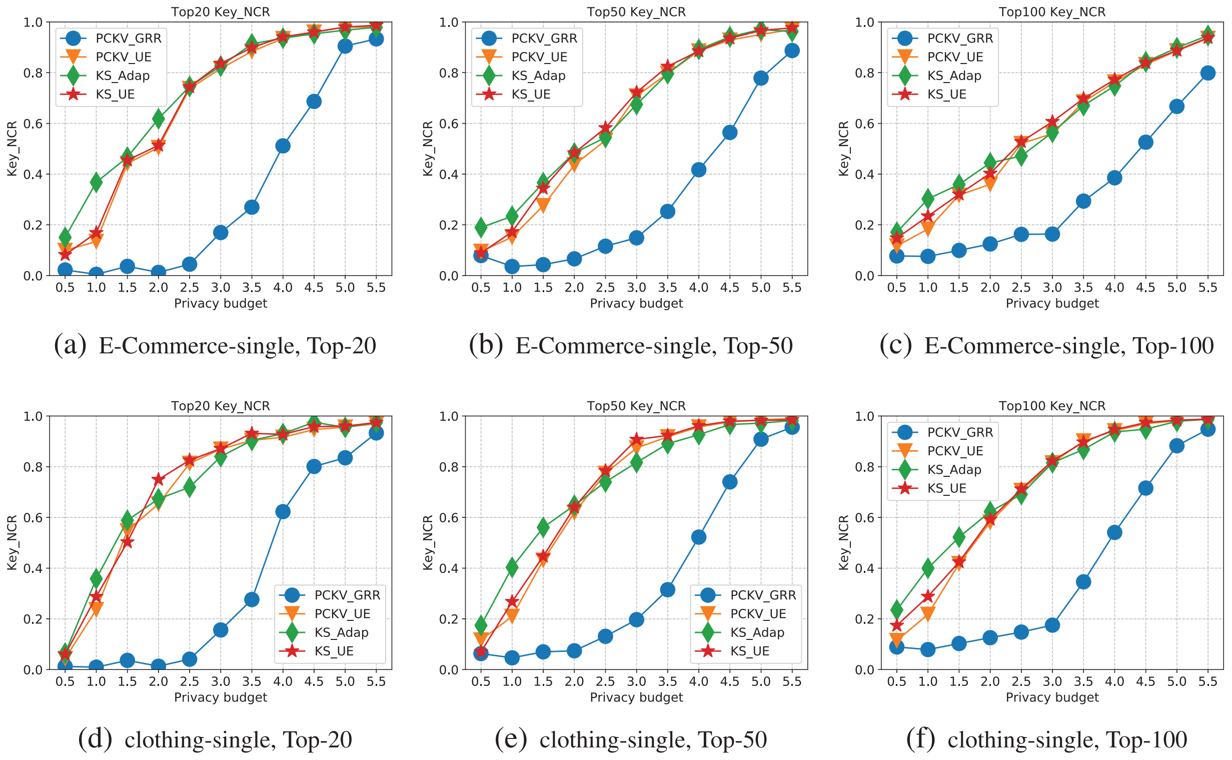
For KS-UE, PCKV-UE and PCKV-GRR, we set $l = 2$ for multiple key-value pairs. For KS-GRR, two groups of users are equally divided for singleton pair.

All experiments were conducted on an Intel Core(TM) i7-6700 3.40 GHz PC with 16 GB RAM. The results are averaged over 10 runs.

Singleton key-value pair. Figs. 3–5 depict the metric evaluation on the E-Commerce and Clothing singleton key-value pair datasets.

[1] https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews

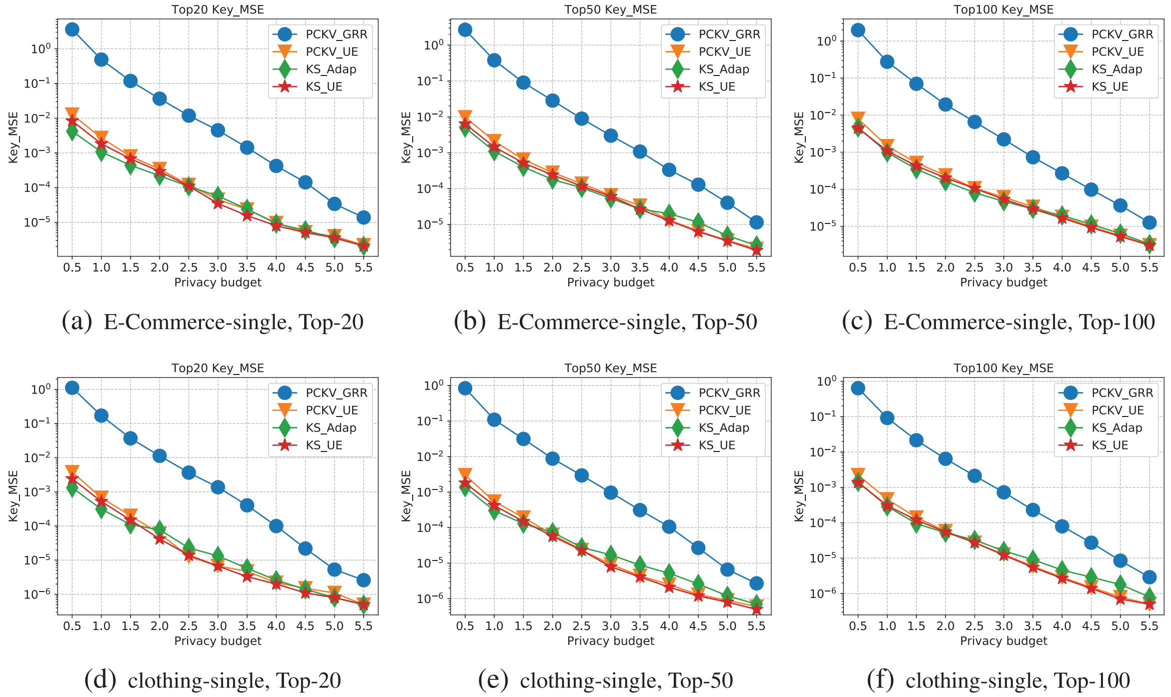
[2] https://www.kaggle.com/rmisra/clothing-fit-dataset-for-size-recommendation



(a) E-Commerce-single, Top-20 (b) E-Commerce-single, Top-50 (c) E-Commerce-single, Top-100

(d) clothing-single, Top-20 (e) clothing-single, Top-50 (f) clothing-single, Top-100

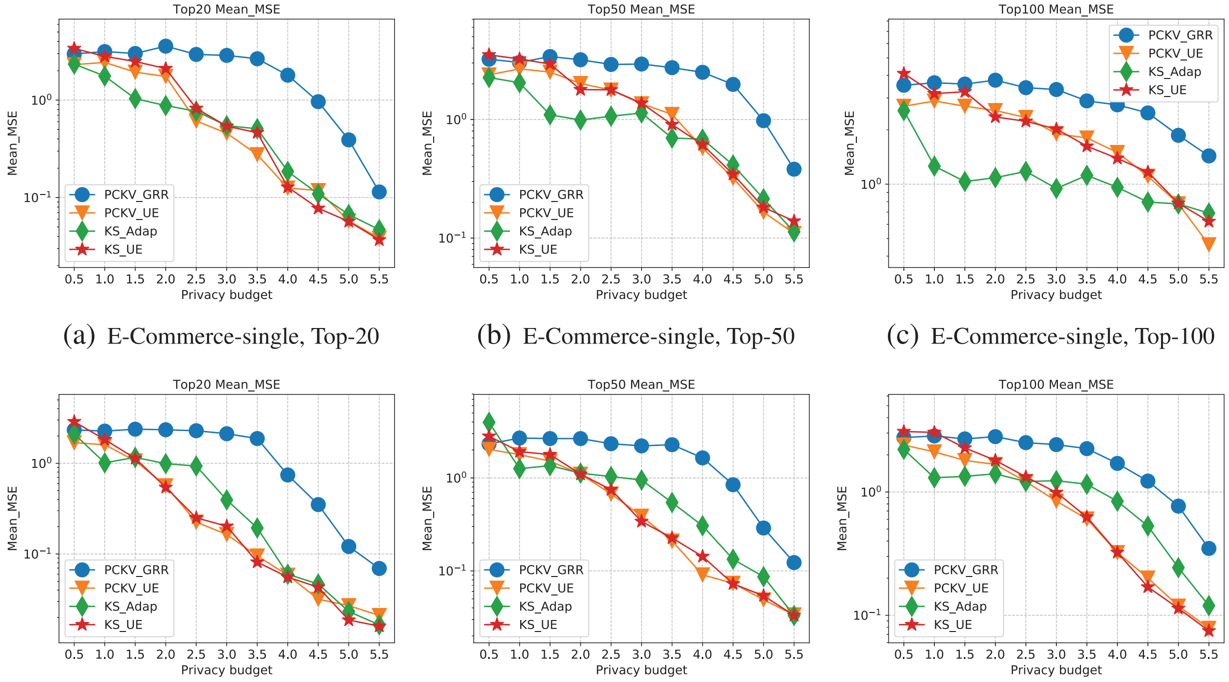
Figure 3: NCR for single dataset



(a) E-Commerce-single, Top-20 (b) E-Commerce-single, Top-50 (c) E-Commerce-single, Top-100

(d) clothing-single, Top-20 (e) clothing-single, Top-50 (f) clothing-single, Top-100

Figure 4: Key-MSE for single dataset



(a) E-Commerce-single, Top-20

(b) E-Commerce-single, Top-50

(c) E-Commerce-single, Top-100

(d) clothing-single, Top-20

(e) clothing-single, Top-50

(f) clothing-single, Top-100

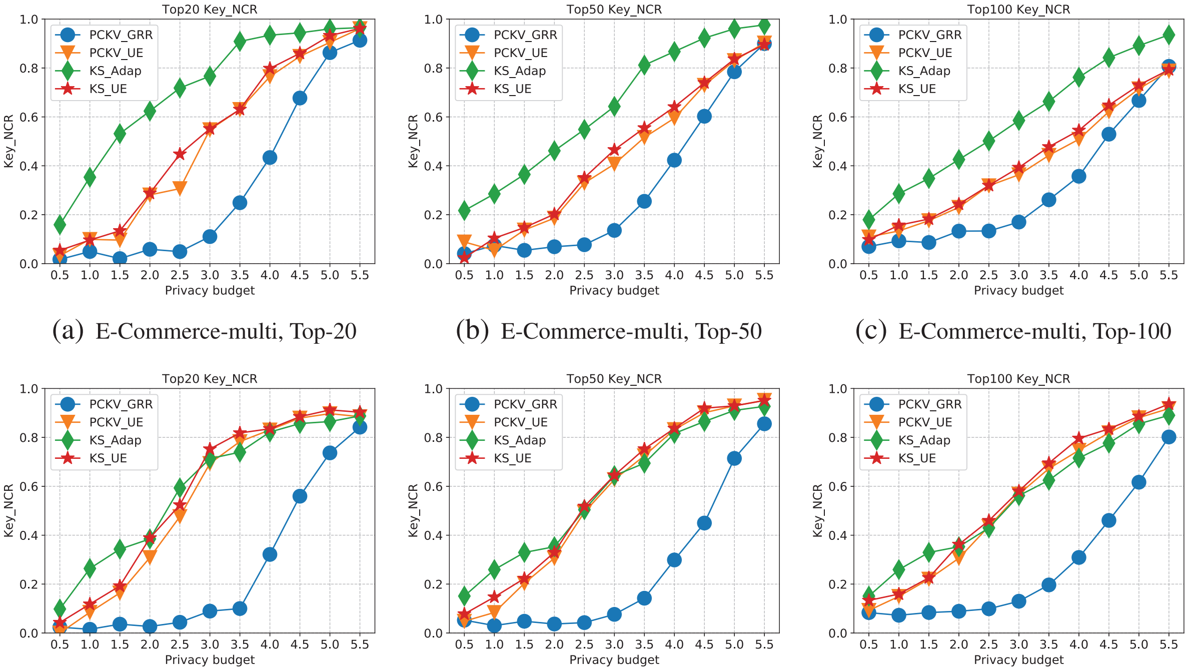
Figure 5: Mean-MSE for single dataset

Firstly, Fig. 3 presents the trendlines of NCR for keys with an increasing privacy budget (ε) when selecting the top 20, 50, and 100 values. As per the NCR definition, higher trendlines correlate with more accurate identification. Our analysis revealed that KS-UE consistently outperforms PCKV-UE. Similarly, KS-GRR notably surpasses PCKV-GRR, exhibiting the best performance at smaller values of ε. Therefore, both KS-UE and KS-GRR demonstrate robust performance in terms of the NCR of $\hat{f}$.

Secondly, Fig. 4 illustrates the trendlines of MSE for keys with an incrementing privacy budget (ε), selecting the top 20, 50, and 100 values. As per the MSE definition, lower trendlines are indicative of higher accuracy. It is observed that KS-UE continuously outstrips PCKV-UE. Moreover, KS-GRR considerably outperforms PCKV-GRR. Consequently, KS-UE and KS-GRR display robust performance regarding the MSE of $\hat{f}$.

Lastly, Fig. 5 demonstrates the trendlines of MSE for the mean values with an increasing privacy budget (ε), while selecting the top 20, 50, and 100 values. The trendlines of KS-UE and PCKV-UE closely intersect. Although KS-UE matches the utility of PCKV-UE, the allocated budget for the value is smaller and a higher NCR implies more identified keys. Therefore, KS-UE exhibits exceptional performance. For $\varepsilon > 3$, the variance of KS-GRR is lower than that of PCKV-GRR. Moreover, for $\varepsilon \geq 6$, KS-GRR even surpasses both PCKV-UE and KS-UE. When combined with the NCR and MSE of $\hat{f}$, the $\hat{m}$ of KS-GRR demonstrates an improvement over PCKV-GRR, though it does not exhibit a significant advantage over PCKV-UE. On the whole, KS-UE and KS-GRR exhibit effectiveness and efficiency, which is consistent with our theoretical results.

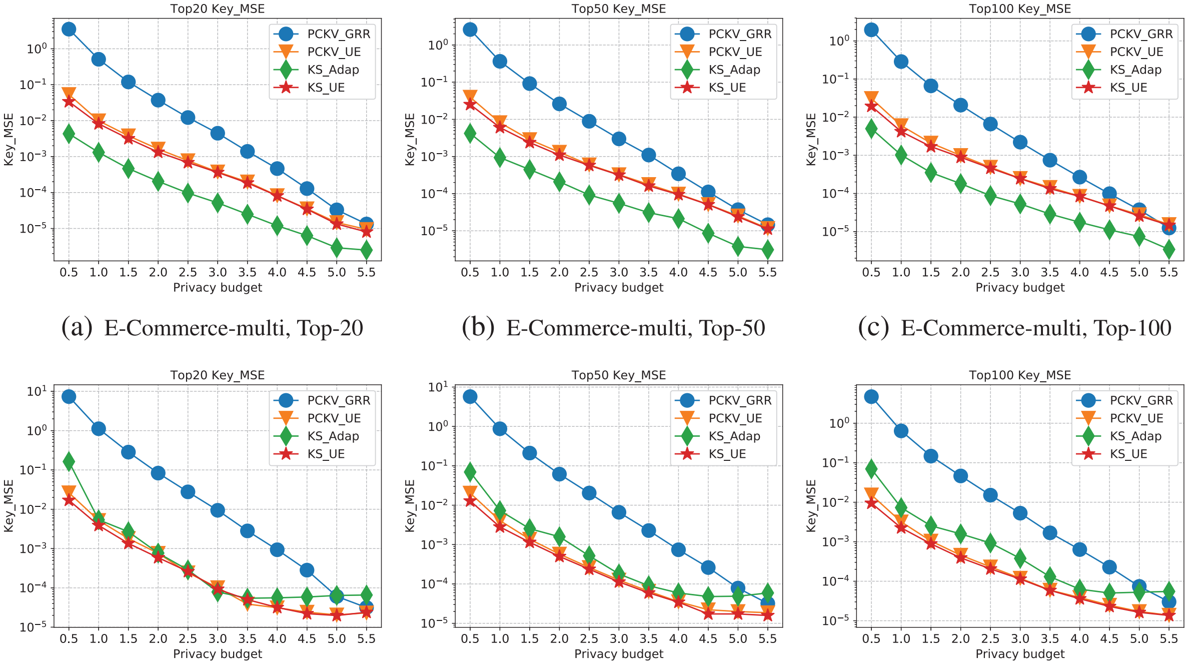
Multiple key-value pairs. Figs. 6–8 present the metric evaluation for the E-Commerce and Clothing datasets featuring multiple key-value pairs.



(a) E-Commerce-multi, Top-20 (b) E-Commerce-multi, Top-50 (c) E-Commerce-multi, Top-100

(d) clothing-multi, Top-20 (e) clothing-multi, Top-50 (f) clothing-multi, Top-100

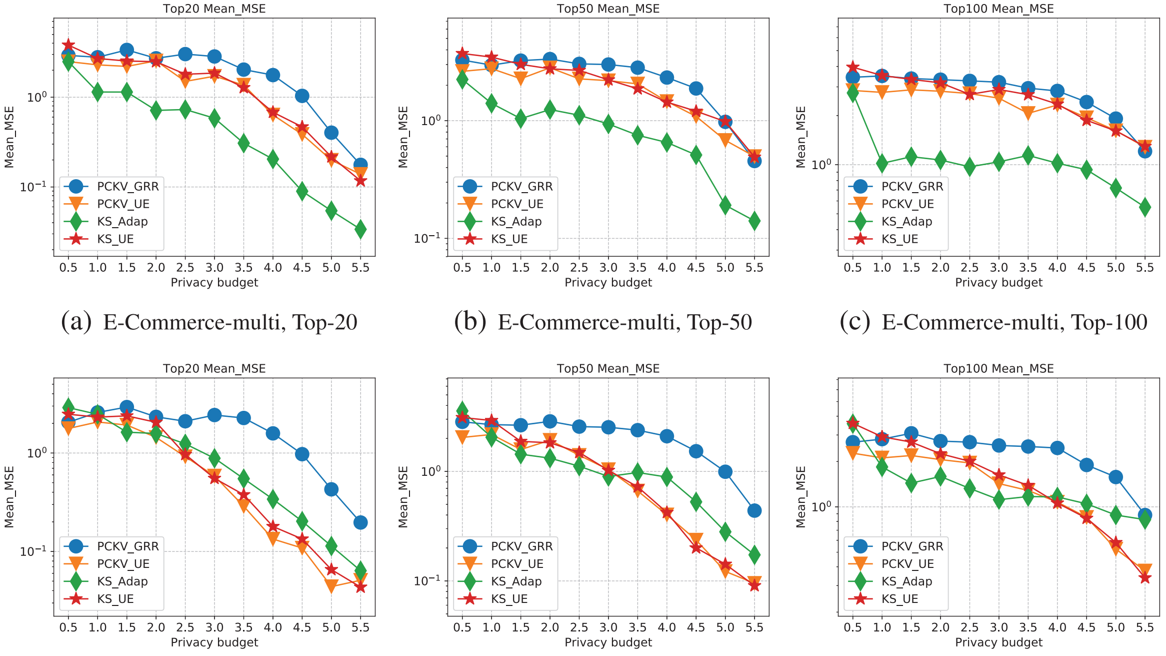
Figure 6: NCR for multiple dataset



(a) E-Commerce-multi, Top-20 (b) E-Commerce-multi, Top-50 (c) E-Commerce-multi, Top-100

(d) clothing-multi, Top-20 (e) clothing-multi, Top-50 (f) clothing-multi, Top-100

Figure 7: Key-MSE for multiple dataset



(a) E-Commerce-multi, Top-20 (b) E-Commerce-multi, Top-50 (c) E-Commerce-multi, Top-100

(d) clothing-multi, Top-20 (e) clothing-multi, Top-50 (f) clothing-multi, Top-100

Figure 8: Mean-MSE for multiple dataset

Firstly, Fig. 6 depicts the trendlines for the NCR of keys, corresponding to the increasing privacy budget (ε), for the top 20, 50, and 100 selected values. By definition, a higher trendline suggests more accurate identification. KS-UE consistently outperforms PCKV-UE, and KS-GRR significantly surpasses PCKV-GRR. In particular, KS-GRR exhibits the best performance across all ε values for the E-Commerce-multiple dataset, and performs optimally under smaller privacy budgets for the clothing-multiple dataset. Hence, both KS-UE and KS-GRR demonstrate excellent performance in terms of the NCR of $\hat{f}$.

Secondly, Fig. 7 illustrates the MSE trendlines for keys, with respect to the incrementing privacy budget (ε), for the top 20, 50, and 100 selected values. In line with the definition of MSE, lower trendlines equate to more accurate identification. Here, KS-UE consistently surpasses PCKV-UE, and KS-GRR considerably outperforms PCKV-GRR, even showing the best performance for large privacy budgets on the E-Commerce-multiple dataset. Thus, both KS-UE and KS-GRR excel in relation to the MSE of $\hat{f}$.

Lastly, Fig. 8 presents the MSE trendlines for the mean values with increasing ε, for the top 20, 50, and 100 selected values. The trendlines of KS-UE and PCKV-UE closely intertwine. Although KS-UE attains equivalent utility with PCKV-UE, the budget allocation for value is smaller and a higher NCR implies more identified keys, signifying excellent performance by KS-UE. KS-GRR generally performs optimally for small ε values. For large ε values, KS-GRR exhibits the best performance for E-Commerce-multiple and is relatively efficient in the clothing-multiple dataset. Combined with the NCR and MSE of $\hat{f}$, the $\hat{m}$ of KS-GRR demonstrates improvement over PCKV-GRR, although it does not hold a significant advantage over PCKV-UE. Overall, KS-UE and KS-GRR both exhibit effectiveness and efficiency, which aligns with our theoretical findings.

Analysis. The experimental outcomes reveal that the KS-UE and KS-GRR algorithms predominantly surpass other algorithms in most situations. The advantages of KS-UE have already been discerned in the error analysis in Section 4.4; its Mean Squared Error (MSE) is lower than prior algorithms, thus reflecting more superior results in the experiments. KS-GRR, by adopting a key-priority strategy, ensures the accuracy in collecting keys and the validity of the values under such keys, once accurate keys have been obtained by collecting the frequency of keys.

6 Conclusions

In this research, we explore methods for key-value data collection under the constraint of local differential privacy to obtain key frequencies and corresponding mean values. We design two innovative strategies, namely, Key Strategy KS-UE and KS-GRR, applicable in interactive and non-interactive frameworks, respectively. Since mean estimation is intrinsically dependent on key frequencies, precise key collection is vitally important not only for estimating key frequencies but also for deriving accurate mean values. KS-UE is specifically designed to tilt the privacy budget towards the key, thereby enhancing the estimation of keys, while KS-GRR implements a group-and-interactive strategy to identify the top frequent keys, subsequently enabling accurate frequency and mean estimation. Empirical validation supports our theoretical analysis, demonstrating the effectiveness of both KS-UE and KS-GRR. In terms of future work, we aim to enhance the efficiency of the candidate domain and reduce both communication and time complexity. These refinements will potentially allow for the broader implementation of these methodologies in practical applications, thereby contributing significantly to the field of privacy protection.

Acknowledgement: This papers logical organisation and content quality have been enhanced, so the authors thank anonymous reviewers and journal editors for assistance.

Funding Statement: This work is supported by a grant from the National Key R&D Program of China.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Dan Zhao, Chuanwen Lu; data collection: Dan Zhao, Yang Liu; analysis and interpretation of results: Dan Zhao, Yang You; draft manuscript preparation: Dan Zhao, Chuanwen Luo, Ting Chen,Yang Liu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data is from https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews and https://www.kaggle.com/rmisra/clothing-fit-dataset-for-size-recommendation.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Chen, R., Li, H., Qin, A., Kasiviswanathan, S. P., Jin, H. (2016). Private spatial data aggregation in the local setting. *2016 IEEE 32nd International Conference on Data Engineering (ICDE)*, pp. 289–300. Helsinki, Finland.
2. Erlingsson, Ú., Pihur, V., Korolova, A. (2014). RAPPOR: Randomized aggregatable privacy-preserving ordinal response. *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1054–1067. Scottsdale, Arizona, USA.

3. Tang, J., Korolova, A., Bai, X., Wang, X., Wang, X. (2017). Privacy loss in apple's implementation of differential privacy on macos 10.12. arXiv preprint arXiv:1709.02753.
4. Ding, B., Kulkarni, J., Yekhanin, S. (2017). Collecting telemetry data privately. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, pp. 3571–3580. Long Beach, CA, USA.
5. Qin, Z., Yang, Y., Yu, T., Khalil, I., Xiao, X. et al. (2016). Heavy hitter estimation over set-valued data with local differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 192–203. Vienna, Austria.
6. Wang, T., Li, N., Jha, S. (2018). Locally differentially private frequent itemset mining. *2018 IEEE Symposium on Security and Privacy (SP)*, pp. 127–143. San Francisco, CA, USA.
7. Zhao, D., Zhao, S., Chen, H., Liu, R., Li, C. et al. (2023). Hadamard encoding based frequent itemset mining under local differential privacy constraints. *Journal of Computer Science and Technology*, *38(6),* 1403–1422.
8. Qin, Z., Yu, T., Yang, Y., Khalil, I., Xiao, X. et al. (2017). Generating synthetic decentralized social graphs with local differential privacy. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 425–438. Dallas, USA.
9. Liu, Y., Zhao, S., Liu, Y., Zhao, D., Chen, H. et al. (2022). Collecting triangle counts with edge relationship local differential privacy. *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pp. 2008–2020. Kuala Lumpur, Malaysia.
10. Ye, Q., Hu, H., Meng, X., Zheng, H. (2019). PrivKV: Key-value data collection with local differential privacy. *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 317–331. San Francisco, CA, USA.
11. Sun, L., Zhao, J., Ye, X., Feng, S., Wang, T. et al. (2019). Conditional analysis for key-value data with local differential privacy. arXiv preprint arXiv:1907.05014.
12. Gu, X., Li, M., Cheng, Y., Xiong, L., Cao, Y. (2020). PCKV: Locally differentially private correlated key-value data collection with optimized utility. *29th USENIX Security Symposium (USENIX Security 20)*, pp. 967–984. Berkeley, CA, USA.
13. Warner, S. L. (1965). Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association, 60(309),* 63–69.
14. Dwork, C. (2008). Differential privacy: A survey of results. *International Conference on Theory and Applications of Models of Computation*, pp. 1–19. Xi'an, China.
15. Dwork, C., Roth, A. (2014). The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science, 9(3–4),* 211–407.
16. Kairouz, P., Bonawitz, K., Ramage, D. (2016). Discrete distribution estimation under local privacy. arXiv preprint arXiv:1602.07387.
17. Bassily, R., Smith, A. (2015). Local, private, efficient protocols for succinct histograms. *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pp. 127–135. Portland, OR, USA.
18. Wang, T., Blocki, J., Li, N., Jha, S. (2017). Locally differentially private protocols for frequency estimation. *Proceedings of the 26th USENIX Security Symposium*, pp. 729–745. Vancouver, BC, Canada.
19. Wang, N., Xiao, X., Yang, Y., Zhao, J., Hui, S. C. et al. (2019). Collecting and analyzing multidimensional data with local differential privacy. *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 638–649. Macau, China.
20. Kulkarni, T., Cormode, G., Srivastava, D. (2017). Marginal release under local differential privacy. arXiv preprint arXiv:1711.02952.
21. Zhang, Z., Wang, T., Li, N., He, S., Chen, J. (2018). CALM: Consistent adaptive local marginal for marginal release under local differential privacy. *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pp. 212–229. Copenhagen, Denmark.

22. Nguyên, T. T., Xiao, X., Yang, Y., Hui, S. C., Shin, H. et al. (2016). Collecting and analyzing data from smart device users with local differential privacy. arXiv preprint arXiv:1606.05053.
23. Ye, Q., Hu, H., Au, M. H., Meng, X., Xiao, X. (2020). Towards locally differentially private generic graph metric estimation. *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pp. 1922–1925. Dallas, TX, USA.
24. Ye, Q., Hu, H., Au, M., Meng, X., Xiao, X. (2020). LF-GDPR: Graph metric estimation with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering (TKDE), 10*, 4905–4920.
25. Cao, Y., Xiao, Y., Xiong, L., Bai, L. (2019). PriSTE: From location privacy to spatiotemporal event privacy. *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1606–1609. Macau, China.
26. Ye, Q., Hu, H., Li, N., Meng, X., Zheng, H. et al. (2021). Beyond value perturbation: Local differential privacy in the temporal setting. *IEEE INFOCOM 2021–IEEE Conference on Computer Communications*, pp. 1–10. Vancouver, BC, Canada.
27. Li, Z., Wang, T., Lopuhaä-Zwakenberg, M., Li, N., Škoric, B. (2020). Estimating numerical distributions under local differential privacy. *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, pp. 621–635. Portland, Oregon, USA.
28. Kulkarni, T. (2019). Answering range queries under local differential privacy. *Proceedings of the 2019 International Conference on Management of Data*, pp. 1832–1834. Amsterdam, Netherlands.
29. Arachchige, P. C. M., Bertok, P., Khalil, I., Liu, D., Camtepe, S. et al. (2019). Local differential privacy for deep learning. *IEEE Internet of Things Journal, 7(7),* 5827–5842.
30. Liu, R., Wu, F., Wu, C., Wang, Y., Cao, Y. et al. (2022). PrivateRec: Differentially private training and serving for federated news recommendation. arXiv preprint arXiv:2204.08146.
31. Chai, Y., Du, L., Qiu, J., Yin, L., Tian, Z. (2022). Dynamic prototype network based on sample adaptation for few-shot malware detection. *IEEE Transactions on Knowledge and Data Engineering, 35(5),* 4754–4766.
32. Qiu, J., Tian, Z., Du, C., Zuo, Q., Su, S. et al. (2020). A survey on access control in the age of internet of things. *IEEE Internet of Things Journal, 7(6),* 4682–4696.
33. Tian, Z., Luo, C., Qiu, J., Du, X., Guizani, M. (2019). A distributed deep learning system for web attack detection on edge devices. *IEEE Transactions on Industrial Informatics, 16(3),* 1963–1971.
34. Ye, Q., Hu, H., Meng, X., Zheng, H., Huang, K. et al. (2023). PrivKVM*: Revisiting key-value statistics estimation with local differential privacy. *IEEE Transactions on Dependable and Secure Computing*, *1,* 17–35.
35. Bassily, R., Nissim, K., Stemmer, U., Thakurta, A. G. (2017). Practical locally private heavy hitters. *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, pp. 2288–2296. Long Beach, CA, USA.
36. Bun, M., Nelson, J., Stemmer, U. (2018). Heavy hitters and the structure of local privacy. *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pp. 435–447. New York, USA.