Computers & Security 151 (2025) 104301

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

Full length article

Multidimensional categorical data collection under shuffled differential privacy

Ning Wang [a], Jian Zhuang [b], Zhigang Wang [a,*], Zhiqiang Wei [b], Yu Gu [c], Peng Tang [d], Ge Yu [c]

[a] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, Guangdong 511442, China
[b] School of Computer Science and Technology, Ocean University of China, Qingdao, Shandong 266100, China
[c] School of Computer Science and Engineering, Northeastern University, Shenyang, Liaoning 110819, China
[d] Key Lab of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, Qingdao, Shandong 266200, China

ARTICLE INFO

Keywords:
Multidimensional categorical data
Amplified privacy
Shuffled differential privacy
Frequency estimation

ABSTRACT

Estimating frequency distributions in multidimensional categorical data is fundamental for many real-world applications, but such data often contains sensitive personal information, necessitating robust privacy protection. The emerging shuffled differential privacy (SDP) model provides a promising solution, yet existing methods are either limited to single-dimensional data or suffer from poor accuracy in multidimensional scenarios. To address these challenges, this paper introduces Multiple Hash Mechanism (MHM), which uses an innovative hash-based local perturbation technique for efficient dimensionality reduction to improve the result accuracy under the SDP framework. Additionally, we provide a detailed analysis of the shuffling benefits of MHM outputs, showing significant accuracy improvements. For cases requiring personalized privacy levels, we propose the Overlapping Group Mechanism, which further enhances the shuffling benefits and boosts overall accuracy. Experimental results on real-world datasets validate the effectiveness of proposed methods.

1. Introduction

The big data era is coming with strong and ever-growing demands on analyzing and collecting personal information to support data analysis tasks. Most of these tasks have a fundamental component relying on frequency distribution among categorical data. For example, in the shopping analysis, the typical frequent itemset mining algorithm, Apriori (Agrawal and Srikant, 1994), checks whether an itemset is frequent by comparing its frequency against a given threshold. Here, an itemset consists of different categorical items from the goods attribute. Another example is the well-known ID3 algorithm (Quinlan, 1986), which constructs a decision tree to classify things with multiple attributes. Frequency distribution is used to compute the information gain so that reasonable attributes can be selected for better classification. The value of each attribute is also categorical, like female vs. male, and overweight or not. However, deriving frequency distributions requires collecting user data, which often contains sensitive personal information, posing significant privacy risks.

To address these risks, privacy protection models like k-anonymity (Caruccio et al., 2022; Aggarwal, 2005; Garg and Torra, 2024) and ϵ-differential privacy (DP) (Li et al., 2017; Niu et al., 2021; Qin et al., 2016) have been developed. The former obscures individual records within groups of k-indistinguishable records to prevent re-identification, but it is vulnerable to attacks such as "homogeneity" (Machanavajjhala et al., 2006) and "background knowledge" (Machanavajjhala et al., 2006). In contrast, differential privacy guarantees that no adversary can confidently infer private values, regardless of their prior knowledge. A recent advancement, shuffled differential privacy (SDP) (Balle et al., 2019; Cheu et al., 2019; Erlingsson et al., 2019), offers enhanced privacy by introducing a shuffler: (1) each user first sends data locally perturbed with noise to a simple yet trusted shuffler; (2) the latter does nothing but permutes received reports before submitting them to the non-trusted curator. Benefiting from the *two procedures*, true values never leave local users and the curator is confused about "who reports what". SDP thereby enables safe data analysis, even though there is no trusted curator. Its key shuffler design clearly brings privacy amplification, which decreases additionally added noise and hence maximizes the utility of statistical information. Considered the superiority of the SDP model, this paper focuses on publishing frequency distribution estimation on multidimensional categorical data under SDP.

Despite the advantages of SDP, most existing solutions are limited to single-dimensional data, where each user contributes values from a single attribute (Balle et al., 2019; Cheu et al., 2019; Erlingsson

* Corresponding author.
E-mail address: wangzhigang1210@163.com (Z. Wang).

https://doi.org/10.1016/j.cose.2024.104301
Received 26 March 2023; Received in revised form 26 November 2024; Accepted 20 December 2024
Available online 8 January 2025
0167-4048/© 2024 Published by Elsevier Ltd.

et al., 2019). However, many real-world applications involve several attributes, i.e., multidimensional data, like attributes used in ID3. By the composition property of DP, we can simply apply existing single-attribute solutions for each dimension, but a great number of noise will be incurred due to the large dimensionality. For better utility, another two representatives, like FLAME (Liu et al., 2021) and TCM (Wang et al., 2021a), are recently proposed with smart dimension reduction policies. The former asks each user to independently sample only one dimension value for information collection, to satisfy the single-SDP constraint. Note that SDP must analyze privacy amplification by the prior-known and deterministic number of collected values (i.e., shuffling participants), then it can compute reasonable noise required in the first procedure to avoid overreaction. FLAME thereby gives a predefined number before sampling and finally achieves the target by enabling shuffler to add dummy pairs, if necessary. However, it is debatable to enhance shuffler's ability, because the new function will challenge the traditionally Trusted Execution Environment (TEE)-based implementation (Bittau et al., 2017). The dummy values also imports extra errors. The latter transforms multi-dimensional data into single-dimensional set-valued data, so that existing techniques can be used. It clearly follows the strict SDP definition. However, the transformation is compute-intensive; worse, users need to coordinate with each other to consistently encode intersected values from different attribute domains, which is communication-intensive. Both of them generate significant performance penalty.

To address these limitations, this paper proposes *Multiple Hash Mechanism* (MHM), a novel SDP-compliant solution for efficiently publishing frequency distributions of multidimensional categorical data. MHM's perturbation method handles the multi-dimension problem head on in two phases. Users first select several hash functions and then project the values from different dimensions into a given domain. The shuffler only receives a single domain-related noisy value and hash functions as input for shuffling. As a result, the number of shuffling participants is fixed and can be easily pre-inferred; and we do not need to care about the same (intersected) values from different dimensions. That clearly eliminates the negative side effect caused by sampling- and transforming-based dimension reduction policies.

Besides, as mentioned above, the privacy amplification of shuffling can improve result utility. Analyzing a tight lower bound of the impact is significantly important, because we can compute reasonable noise locally added by users—which is certainly as small as possible but should be big enough to satisfy the privacy protection requirement. Thus, such analysis is always a hot research topic for SDP. Most of existing efforts give the bound based on the traditional DP definition (Balle et al., 2019; Wang et al., 2020; Li et al., 2020; Wang et al., 2021a). That simplifies the analysis but generates a loose bound, which has a negative impact on accuracy and enforces users to add excessive noises. Inspired by the privacy amplification analysis based on the relationship between divergences and DP (Balle et al., 2019), MHM also makes a strict bound analysis for shuffling. The divergence is essentially related to a sole probability mass function used in adding local noise. MHM challenges this constraint, because its phased perturbation generates not only traditional perturbed values (Balle et al., 2019) but also indexes of hash functions, both of which follow two different probability mass functions. Thus, a new function is constructed on top of integrating the two separate ones, and its key properties are extracted to derive a strict privacy benefit bound.

Finally, we also extend our proposal MHM to the setting where the users are with personalized privacy protection requirement. Many pioneers have explored solutions for traditional DP (Jorgensen et al., 2015; Chen et al., 2016), but few efforts are devoted into the new SDP. Our MHM is also designed with the uniform assumption. However, we can group users by their budgets and then invoke MHM multiple times for every group. The grouping criterion is then becoming another key issue. Because privacy amplification in SDP is very sensitive to the number of shuffling participants, we are motivated to maximize the group size to enhance the amplification benefit. Towards this end, we propose an *Overlapping Group Mechanism* where each privacy budget is further cut into several splits as grouping criterions, and then the user with a large budget can participate in shuffling across different groups with high priority. That clearly increases the size of each group. To further boost the benefit, we particularly study the important issue about how to heuristically split budgets. The result utility is thereby significantly improved.

We now summarize the major contributions as below.

(1) We propose a new SDP-compliant mechanism MHM involving a novel local perturbation mechanism to publish the frequency distribution estimation on the multidimensional categorical data.

(2) We give a strict privacy benefit bound of shuffling involved in MHM, by resorting to the relationship between divergences and differential privacy, which can improve the result utility with the same privacy budget setting.

(3) We study SDP with personalized privacy budgets and put forward a new grouping policy *overlapping group mechanism* to enhance the result utility.

(4) We conduct extensive experiments over a broad spectrum of real datasets to demonstrate that our proposals significantly outperform the state-of-the-art solutions.

The remainder of this paper is organized as follows. Section 2 reviews existing studies about the problems this paper focuses on. Section 3 presents preliminaries about differential privacy and the existing methods. Section 4 gives the detailed design of MHM. Section 5 proposes *overlapping group mechanism* for shuffler-assisted personalized differential privacy. Section 6 evaluates the usefulness of our proposals, and Section 7 finally concludes the paper.

2. Related work

Differential privacy (DP) contains three representative models with different settings, centralized differential privacy(CDP), local differential privacy(LDP) and shuffled differential privacy(SDP). In CDP, the trusted curator collects users' data and adds noise to the aggregated result (Dwork et al., 2006b). Differently, LDP works with an untrusted curator. Thus, every user perturbs her/his own data locally and then sends privatized data to the curator (Fanti et al., 2016). It provides a stronger privacy assurance to users, as true values never leave local devices. However, the accumulated noise scale is very high and hence decreases result utility. To improve utility, SDP (Balle et al., 2019; Cheu et al., 2019; Ghazi et al., 2020) employs the Encode-Shuffle-Analyze architecture (Bittau et al., 2017), to balance utility and privacy protection level. It adds a trusted shuffler, so as to anonymize messages collected from users before they are sent to the curator, and hence achieves privacy amplification (Bittau et al., 2017).

For the frequency distribution estimations problem, most of SDP-related studies (Balle et al., 2019; Cheu et al., 2019; Erlingsson et al., 2019; Feldman et al., 2021) focus on the one with a single categorical attribute. Among them, Ref. Cheu et al. (2019) proposes to collect data from users with a binary categorical attribute. Ref. Erlingsson et al. (2019) allows flexible interoperation between users and shuffler. Balle et al. (2019) put forward the privacy blanket technique to process the problem with arbitrary domain size. It especially establishes two lines to derive the privacy amplification bound: one is based on the traditional DP definition and the other resorts to the relationship between divergences and DP. Many works (Wang et al., 2020; Li et al., 2020; Wang et al., 2021a) follow the first line with different local randomizers, including traditional OLH (Wang et al., 2020), CM (Wang et al., 2021a), and the newly proposed protocols (Li et al., 2020). For example, Li et al. (2020) propose the dummy blanket technique, in which each user generates some dummy values except for the perturbed version of her own true value. And these dummy values also participate in the shuffling phase. To some extent, the dummy values can hide the true information, which brings privacy amplification. Unfortunately,

Table 1
Summary of important related works.

Category	Existing works	Techniques	Shortcomings
LDP + single dimension	Wang et al. (2017)	Encode the value into one-hot vector or hash the value into a smaller domain, and then perturb it for publication.	They just can publish the frequency estimation on a single dimension.
LDP + multidimensional	Wang et al. (2019b) and Liu et al. (2024)	Choose a dimension or traverse each dimension, and then publish the value from the dimension by using the one dimensional publication technique.	They cannot be directly used for the SDP setting.
SDP + single dimension	Balle et al. (2019), Cheu et al. (2019), Erlingsson et al. (2019), Feldman et al. (2021), Wang et al. (2020), Li et al. (2020) and Wang et al. (2021a)	Invoke the LDP-compliant perturbation protocol as the local randomizer and design privacy amplification analysis strategy for shuffling.	They cannot handle the multidimensional frequency distribution estimation and the privacy amplification analysis cannot be directly used for the multidimensional setting.
SDP + multidimension	Scott et al. (2022) and Liu et al. (2021)	Invoke the protocols proposed for LDP-compliant multidimensional frequency estimation as the local randomizer and directly use the privacy amplification conclusion derived in the setting of a single dimension.	They ignore the negative impact of non-deterministic shuffling participants caused by sampling or incur extra communication cost.

this line requires that the output distribution of the local randomizer can be decomposed into a uniform distribution and another distribution expressed by an indicator function for true values. In contrast, the second line is universal yet complicated. It needs to use some property parameters of output distributions, which leads to a thorough analysis required. Besides, Balle et al. (2019) have shown that the second line can derive a much better lower bound for privacy amplification. Although the above works proposed for a single categorical attribute cannot be used to solve our problem for multidimensional attributes, the privacy amplification analysis techniques from Ref. Balle et al. (2019) can inspire us to analyze the privacy amplification of shuffling in the multidimensional setting.

There exist several works about multidimensional data publication under LDP. They choose a dimension (Wang et al., 2019b) or traverse each dimension (Liu et al., 2024), and then publish the value from the dimension by using the one-dimensional publication technique, such as OUE (Wang et al., 2017), OLH (Wang et al., 2017) or PM (Wang et al., 2019b). But these works under LDP cannot be directly used for the SDP setting. As far as we know, there exist only three SDP-based works (Scott et al., 2022; Liu et al., 2021; Wang et al., 2021a) targeting at our problem. Their common idea is to reduce the multidimensional user data into one dimension (attribute) and then invoke the single-attribute-SDP techniques discussed above. Two of them resort to sampling-based reduction, which is originally proposed in Section 4.B of Ref. Wang et al. (2019b) for collecting multidimensional data under LDP and also used in the following-up work (Wang et al., 2021b). Scott et al. (2022) analyze the privacy amplification in a statistical way, which ignores the negative impact of non-deterministic shuffling participants caused by sampling. Liu et al. (2021) propose FLAME to remedy this problem but complicates the shuffler design. Instead of sampling, Wang et al. (2021a) propose TCM to transform the multidimensional data into the multiple values from a single attribute (dimension) directly, but incurs heavy performance penalty. We will show more details about FLAME and TCM in Section 3.3 since they are the most related works. Table 1 classifies some important related works about frequency distribution estimation in the LDP and SDP settings to better distinguish them from ours.

We are aware that our problem also equivalently exists when answering the k-way marginal query and the range query. These queries are answered by estimating frequency distribution on decomposed multiple sketches which can be regarded as our dimensions. Existing related works (Zhang et al., 2018; Ren et al., 2018; Wang et al., 2022, 2019a; Yang et al., 2020; Du et al., 2021) mainly focus on bridging top-level queries and bottom-level estimations. After that, they just simply use LDP-compliant techniques for estimation, which is the focus of this paper but we use the advanced SDP. Clearly, their technical contributions are completely orthogonal to ours. In fact, as analyzed above, our SDP-compliant component can be plugged into their frameworks to replace the built-in LDP-compliant component, so as to further boost answer utility. Our experiments have evaluated the performance of our proposal on these queries.

Last, we outline existing works about personalized privacy expectations. Pioneering works are related to CDP. Jorgensen et al. (2015) use non-uniform sampling and exponential mechanism to publish data, where the sampling probability and quality function are customized by personalized privacy budgets. Li et al. (2017) propose to group user data based on the different privacy preferences and then apply a DP algorithm for each group. Niu et al. (2021) design an iterative framework to publish data without privacy budget waste. There also exist some works under LDP. Chen et al. (2016) design an efficient personalized count estimation protocol. Nie et al. (2019) design a recycle and combination framework for histogram estimation. Liu et al. (2024) target at the limited personalized privacy scenario, where some attribute values are sensitive and some are not. They improve the traditional OUE technique (Wang et al., 2017), in order to publish the value from this kind of attributes with high utility. As far as we know, there is no work under personalized SDP. Compared against CDP and LDP, it is more difficult for SDP because the privacy amplification of shuffling is sensitive to the number of participants, which complicates the policy design for personalized privacy budgets.

3. Preliminaries

This section introduce necessary background knowledge about differential privacy (Section 3.1), our problem definition (Section 3.2) and the most representative related works (Section 3.3).

3.1. Differential privacy

The CDP setting involves a trusted curator and a number of (say, n) users, each of which, u_i possesses a data record x_i containing private information. Since the curator is trustworthy, she has access to n true data records directly. Thus, given a query, the curator can safely publish the result of this query by perturbing the true answer with CDP, which is defined as follows.

Definition 1 (*Centralized Differential Privacy, CDP Dwork et al., 2006a*)**.** A randomized function f satisfies (ϵ, δ)-DP, where $\epsilon, \delta \geq 0$, if and only if for any two neighboring datasets D and D', and for any set O of possible outputs of f, we have

$$\Pr[f(D) \in O] \leq e^{\epsilon} \Pr[f(D') \in O] + \delta.$$

We say two datasets are neighboring if and only if they differ in only one tuple. ϵ is called privacy budget, which controls the strength of privacy protection. A smaller budget means stricter privacy protection. When $\delta \geq 0$, (ϵ, δ)-DP is termed as approximate DP. When $\delta = 0$, we simplify the notation as ϵ-DP and call it as pure DP. In this paper, we are referring to both of them when saying DP; and if necessary, we

distinguish them by the existence of the parameter δ. Considering the different protection expectations of users, we then give the personalized CDP definition.

Definition 2 (*Personalized CDP Jorgensen et al., 2015*)**.** For a personalized privacy budget $S = \{(\epsilon_1, \delta_1), \ldots, (\epsilon_n, \delta_n)\}$ and dataset D consisting of data from a set of users $U = \{u_1, \ldots, u_n\}$, where u_i is with budget (ϵ_i, δ_i), a randomized function f satisfies S-personalized DP, if and only if for any two neighboring datasets D and D' which differ in one arbitrary user u_i, and for any set O of possible outputs of f, we have

$$\Pr[f(D) \in O] \leq e^{\epsilon_i} \Pr[f(D') \in O] + \delta_i.$$

However, the trusted curator assumption in CDP is too strong to be held true in many real scenarios. To protect privacy, records cannot leave the user side. Thus, each user has to perturb her data locally before sending it to the non-trusted curator. The latter answers queries based on noisy information. The following defined LDP model can handle this scenario.

Definition 3 (*Local Differential Privacy, LDP Erlingsson et al., 2014*)**.** A randomized function f satisfies (ϵ, δ)-LDP if and only if for any two tuples x_i and x_i', and for any set O of possible outputs of f, we have

$$\Pr[f(x_i) \in O] \leq e^{\epsilon} \Pr[f(x_i') \in O] + \delta.$$

Like ϵ-DP, most of LDP works focus on the case with $\delta = 0$, i.e., ϵ-LDP. Since LDP answers the query based on n noisy reports, a lot of noise will be incurred and then yield accuracy penalty. For noise reduction, a new model is proposed with an additional shuffling component, termed as *Shuffled Differential Privacy* (SDP) (Balle et al., 2019).

Compared with LDP, the biggest difference in SDP is a new trusted shuffler between users and the non-trusted curator. It permutes noisy reports once they are all received from n users and hence makes them anonymous to the curator. Now the adversary cannot link a specific user to her report, which brings privacy amplification. The noisy reports are generated by perturbing user data through a local randomizer $R : X \to Y$, resembling LDP. Here X is the raw record domain and Y is the perturbed domain. A function $S : Y^n \to Y^n$ is then used to permute/shuffle n record reports.

Suppose R in LDP satisfies ϵ_l-LDP and $M = S \circ R : X^n \to Y^n$ in SDP satisfies (ϵ_s, δ_s)-DP. (ϵ_s, δ_s) is guaranteed by adding noise in the first phase and shuffling in the second phase; while, ϵ_l is guaranteed solely by adding noise. LDP as a component actually works in the first phase of SDP. Undoubtedly, when adding the same noise, benefitting from privacy amplification in the second phase, the output of M in SDP can provide much more stricter privacy protection than the one of the sole R in LDP, i.e., $\epsilon_s < \epsilon_l$. In reverse, from the perspective of end-users, give the same protection requirement, the LDP-component in SDP adds small noise, and hence can improve the accuracy of final results.

Such a shuffling design in SDP is simple yet effective, but a new challenge is how to exactly evaluate its amplified privacy. If a strictly lower bound is known, then SDP can add noise as small as possible for the LDP component in the first phase.

Nowadays, the up-to-date privacy amplification analysis works (Wang et al., 2020; Li et al., 2020; Wang et al., 2021a) use the *privacy blanket* technique (Balle et al., 2019). Its core idea is to decompose the local randomizer R into two parts: with probability γ it reports a value following a probability mass function (pmf) υ_{x_i} dependent on the true value $x_i \in X$; with probability $1-\gamma$ it reports a uniformly random value following pmf ω independent of x_i, where ω is referred to as privacy blanket distribution. So the pmf with true data x_i of R can be written as

$$r_{x_i} = (1-\gamma)\upsilon_{x_i} + \gamma\omega.$$

Obviously, the privacy blanket distribution can hide the reports from υ_{x_i} to some extent. *Privacy amplification random variable* is used to quantify the level of hiding, which is defined as follows:

$$L_{\epsilon_s}^{x_i, x_i'} = \frac{r_{x_i}(W) - e^{\epsilon_s} r_{x_i'}(W)}{\omega(W)}$$

where $W \sim \omega$ is a random variable sampled from ω.

By Ref. Balle et al. (2019), based on properties of $L_{\epsilon_s}^{x_i, x_i'}$, Lemma 1 gives the hockey-stick divergence of order e^{ϵ_s} between $M(D)$ and $M(D')$ on any pair of neighboring datasets D and D'.

Lemma 1. *Given fixed $\epsilon_s \geq 0$ and any pair of neighboring datasets $D, D' \in X^n$ differing in the nth user's data, i.e., $x_n \neq x_n'$, let $L_{\epsilon_s}^{x_n, x_n'}$ be the privacy amplification random variable with $E L_{\epsilon_s}^{x_n, x_n'} = 1 - e^{\epsilon_s} = -a \leq 0$, $E(L_{\epsilon_s}^{x_n, x_n'})^2 \leq c$ and $b_- \leq L_{\epsilon_s}^{x_n, x_n'} \leq b_+$. γ is the probability to report a value from the pdf of ω. Then the hockey-stick divergence of order e^{ϵ_s} between $M(D)$ and $M(D')$ is*

$$D_{e^{\epsilon_s}}(M(D) \parallel M(D')) \leq \frac{1}{\gamma n} \sum_{m=1}^{n} \binom{n}{m} \gamma^m (1-\gamma)^{n-m} \beta(a, b, c, m)$$

where $\beta(a, b, c, m) = \frac{b_+}{am \log\left(1 + \frac{ab_+}{c}\right)} e^{\frac{-mc}{b_+^2} \phi\left(\frac{ab_+}{c}\right)}$, *and* $\phi(x) = (1+x)\log(1+x) - x$.

Lemma 2 (Barthe and Olmedo, 2013) shows the relationship between hockey-stick divergence and (ϵ_s, δ_s)- DP.

Lemma 2. *A mechanism $M : X^n \to Y$ is (ϵ, δ)-DP if and only if $D_{e^{\epsilon}}(M(D) \parallel M(D')) \leq \delta$ for any neighboring datasets D and D'.*

Accordingly, to infer amplified privacy, we can analyze the pmf used in R to derive ϵ_l-related expressions about parameters a, b_-, b_+ and c. Together with Lemmas 1 and 2, then we can get the relationship between ϵ_l with ϵ_s and δ_s. Given ϵ_l (ϵ_s) and δ_s, ϵ_s (ϵ_l) is derived.

3.2. Problem definition

This paper focuses on frequency distribution estimations on multidimensional categorical data under SDP. Each user first invokes an LDP component to perturb her own private data and then reports a noisy version to the shuffler. Once all n reports are received, the shuffler permutes them before sending them to the non-trusted curator. The latter finally executes the frequency estimation task to answer queries. Now our goal is to maximize the accuracy of answers, while satisfying (ϵ_s, δ_s)-DP privacy protection required by end-users.

More formally, each user u_i's private record data x_i contains d categorical attributes A_1, A_2, ..., and A_d, and x_{ij} denotes the value of A_j. Let C_i indicate the domain of A_i ($1 \leq i \leq d$). Without loss of generality, we assume that attribute A_i with $|C_i|$ distinct values has a discrete domain $C_i = \{1, 2, \ldots, |C_i|\}$. Thus, the curator actually performs frequency estimation on the domains of d attributes. Let F be the true frequency distribution, in which F_k and F_{kj} are respectively associated with A_k and the specific value j within it, i.e., $F_{kj} = \frac{|\{u_i | x_{ik} = j\}|}{n}$. We evaluate the accuracy by comparing the derived noisy estimation against F.

3.3. Existing methods

Since there are not SDP-based methodologies which can be directly applied for solving our multidimensional categorical problem, we try to use the variants of existing single-dimension SDP solutions to cope with encountered challenges. We next overview the sampling and the transforming representatives.

FLAME: sampling mechanism with dummy padding (Liu et al., 2021). FLAME asks each user u_i to randomly sample a dimension index s_i from $[d]$, so as to reduce multiple dimensions into a single one. u_i then perturbs the value in s_i as y_i^* using the Randomized Response technique (Warner, 1965) with privacy budget $\epsilon_{l s_i}$. The resulting pair (s_i, y_i^*) is sent to the shuffler. Once the latter receives reports from all

n users, she counts the numbers of reports related to each dimension. Such numbers are clearly non-deterministic because of the random sampling. That challenges the analysis of privacy amplification since a bounded-size database is required (Balle et al., 2019). FLAME thereby assumes a uniform number N_p which is much greater than the average $\frac{n}{d}$ and typically set as $\frac{n}{3}$ by default. If the real number related to some dimension does not reach the pre-assumed threshold, the shuffler should add some dummy pairs, each of which consists of this dimension index and a random value from its domain. Following that, the shuffler permutes $d \cdot N_p$ pairs and then sends them to the curator, to compute the frequency estimation.

Let the final published result satisfy (ϵ_s, δ_s)-DP. The budget ϵ_{lj} used for the jth dimension can be computed by $\log\left(\frac{(N_p-1)\epsilon_{ck}^2}{14k\log(2/(d\delta_s))} - k + 1\right)$, where $\epsilon_{ck} = \log(d(e^{\frac{\epsilon_s}{2}} - 1) + 1)$ and k denotes the domain size of this dimension. The relationship between ϵ_{lj} and (ϵ_s, δ_s) is derived by considering the composition property of DP (McSherry, 2009), and the privacy amplification effectiveness for sampling (Balle et al., 2018) and shuffling (Balle et al., 2019).

Traditionally, the shuffler in SDP is intelligent-unable, which can do nothing except shuffling. The additional counting and adding functions in FLAME poses new implementation challenges in practice. Besides, dummy pairs also incurs errors.

TCM: collision mechanism with transformation (Wang et al., 2021a). Since TCM is a variant of the collision mechanism (CM), we first introduce the latter. CM can publish SDP-compliant frequency distribution estimation on a single categorical attribute. Each user u_i processes m categorical values from attribute A_j. Let $x_i = \{x_{i1}, x_{i2}, \ldots, x_{im}\}$ be the record of u_i, where $x_{ik} \in C_j$ and C_j denotes this attribute domain. Like FLAME, u_i also reports one pair to the shuffler but with privacy budget $\epsilon_l = \log\left(\frac{\epsilon^2(n-1)}{14m\log(2/\delta)} - \frac{g}{m} + 1\right)$ and different pair value. In particular, u_i firstly chooses a function h_i from the universal hash function family $\mathcal{H}$, and then sequentially hashes elements in x_i into m values (denoted by $\mathbf{y}_i$) in $[g]$, where $g = 2m - 1 + me^{\epsilon_l}$. Note that $|\mathbf{y}(x_i)|$ may be less than m due to hash collision. The reported value is from $[g]$, and candidates in $\mathbf{y}_i \subseteq [g]$ should be selected in high probability, which is guaranteed by the following selection probability distribution, where $p = \frac{e^{\epsilon_l}}{me^{\epsilon_l}+g-m}$:

$$\Pr(y_i^* = j) = \begin{cases} p, & \text{if } j \in \mathbf{y}(x_i), \\ \frac{1-|\mathbf{y}(x_i)|p}{g-|\mathbf{y}(x_i)|}, & \text{if } j \notin \mathbf{y}(x_i). \end{cases} \tag{1}$$

The pair value is then $\langle h_i, y_i^* \rangle$ consisting of the used hash function and perturbed value. As usual, the shuffler permutes received reports and then sends them to the curator. In particular, the latter employs a vector F_j^* with size $|C_j|$ to record the frequency distribution on A_j. More specifically, for $\langle h_i, y_i^* \rangle$, if $h_i(k) = y_i^*$, F_{jk}^* increases by 1. The curator regards $\frac{F_{j\cdot}^* - n/g}{p - 1/g}$ as the unbiased frequency estimation.

CM is only suitable for collecting multi-values from a single attribute, instead of multi-attributes. To fully utilize CM, for each user u_i, we can encode her values from multi-attributes as new items and then transform them into the set x_i. In another word, these items are regarded as multi values from some virtual attribute. Then CM can be utilized for data collection. For d dimensions, the total number of such items is $\sum_{i=1}^{d} |C_i|$. We term this CM variant as TCM. However, the raw values from different dimensions should have distinct physical meanings but their expression might be the same. For example, there exist two attributes gender and marital status. Both might have "M" in their domains but the meaning is very different, i.e., it indicates "male" for the former and "married" for the latter. This essentially requires the curator to make an encoding rule and then broadcast it to all users. Then we can guarantee that distributed users can consistently encode such values as distinctly different items. As a side-effect, additional communication costs (broadcasting rule) and computation costs (encoding) are incurred, compared with the original CM.

4. Multidimensional categorical data collection under SDP

This section introduces a new method termed as multiple hash mechanism (MHM) for multidimensional categorical data collection under SDP, without padding or encoding operations. Section 4.1 describes the procedure of this mechanism, and Section 4.2 does a thorough analysis of privacy amplification.

4.1. Multiple hash mechanism

As introduced in Section 3.3, although the function of the shuffler involved in TCM is consistent with that in the traditional SDP model, it generates additional efficiency penalty from perspectives of communication and computation, due to the extra encoding operation. This section thereby designs a new strategy termed as multiple hash mechanism based on TCM. It preserves accuracy but eliminates the complex encoding step, in order to improve the efficiency of TCM.

As shown in Fig. 1, the framework of MHM which guarantees (ϵ_s, δ_s)-DP involves three parts, n users, a curator and a shuffler. On the user side of user u_i, she invokes the local randomizer component with privacy budget ϵ_l to perturb her data and sends the perturbed values to the shuffler. And once the latter receives the messages from all users, she permutes them and sends the permuted results to the curator. Since the shuffler brings privacy amplification, to make the permutated results from the shuffler guarantee (ϵ_s, δ_s)-DP, privacy budget ϵ_l bigger than ϵ_s is used to perturb the data on the user side. And we will illustrate how to compute ϵ_l in Section 4.2. Note that the function of the shuffler in MHM is consistent with the one in the traditional SDP model, which is just responsible for permuting the reported messages from users. Following that, the curator does frequency distribution estimation on her received messages. Compared with the simple and generalized operation brought by the shuffler, the ones implemented on the user side and curator side are customized for our application, which are elaborated detailedly in the following.

Operations on the user side. The key idea of MHM is that each user u_i adopts d hash functions to respectively hash her d data in x_i to the domain $[g]$, instead of using one hash function in TCM. In this way, the same value from different dimensions can be hashed into two different values in $[g]$ with high probability, so that the same value can be distinguishable. On the other hand, since d values are all hashed into $[g]$, each one in $[g]$ carries the information from all dimensions. As a result of that, publishing the hash value contributes to improving accuracy, due to consuming a part of privacy budget and achieving the integrated information of multiple dimensions. Algorithm 1 shows how MHM works on the user side. In particular, as shown in Fig. 1, on the user side of u_i with true data $x_i = (x_{i1}, x_{i2}, \ldots, x_{id})$, for any dimension j, u_i samples a hash function h_{ij} from the hash function family $\mathcal{H}$ and computes the hash value $y_{ij} = h_{ij}(x_{ij})$. Let HR be a set storing the hash values of all dimensions and ℓ be the size of HR. Note that ℓ may be smaller than d, since the values from different dimensions may be hashed into one value. Then we borrow the probability density function in Eq. (1) used in CM to publish a value on $[g]$, which is implemented in Lines 7–11 of Algorithm 1. Then u_i reports the perturbed value as well as d hash functions to curator.

Note that MHM uses multi-hash functions. It is inefficient to transmit d indexes. Here, we design a trick, i.e., each user just samples a hash function h_i with index i in the hash function family for the first dimension and allocates the hash function with index h_{i+j-1} for the jth dimension directly. In this way, it is enough to transmit the information of hash function for the first dimension. In the following, we use MHM to indicate the version with the communication optimization.

Afterwards, u_i sends the pair $\langle h_i, y_i^* \rangle$ consisting of the used hash function and perturbed value to the shuffler. Once the shuffler receives the reported messages from n users, she permutes them and sends them to the curator.

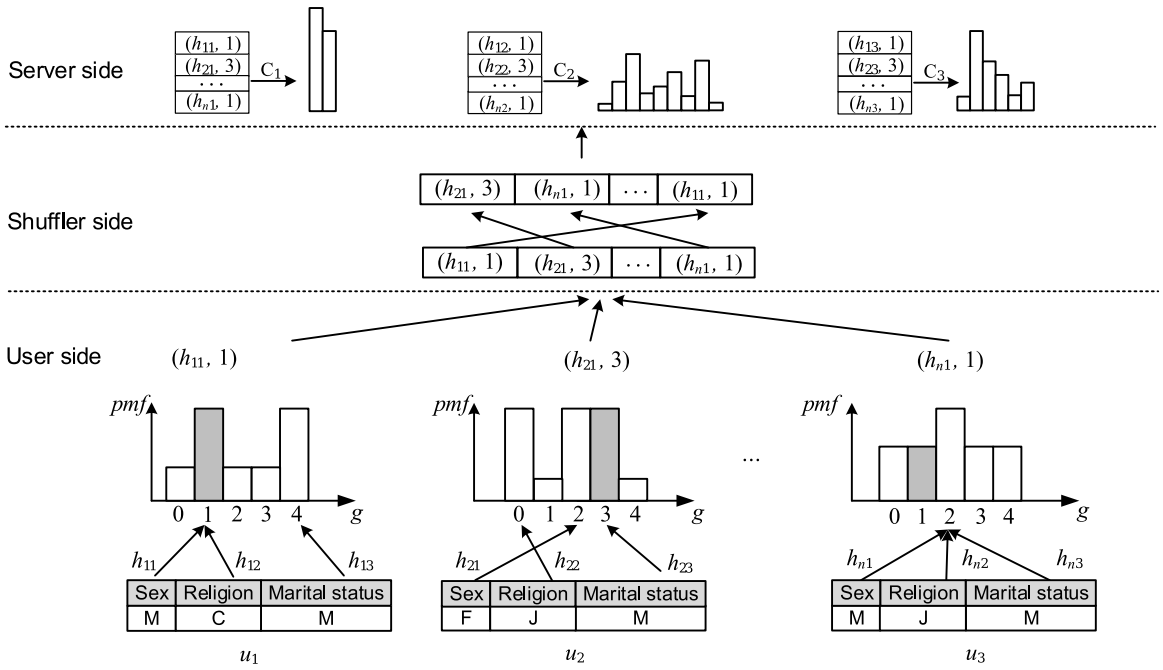

Fig. 1. The framework of MHM.

Algorithm 1: MHM on the user side

Input : True data $x_i = (x_{i1}, x_{i2}, ..., x_{id})$, hash function family $\mathcal{H}$, privacy budget ϵ.

Output: d random hash functions $(h_{i1}, h_{i2}, ..., h_{id})$ and a perturbed value y_i^*.

```
HR ={};
for j from 1 to d do
    Choose a random hash function h_ij from H;
    Use h_ij to hash the jth value x_ij, i.e., y_ij = h_ij(x_ij);
    HR← HR∪{y_ij};
ℓ = |HR|;
Sample ur uniformly at random from [0, 1];
if ur ≤ ℓ · p then
    Sample y_i^* uniformly from HR;
else
    Sample y_i^* uniformly from [g]/HR;
return (h_i1, h_i2, ..., h_id, y_i^*);
```

Algorithm 2: MHM on the curator side

Input : Reported messages $(h_{i1}, h_{i2}, ..., h_{id}, y_i^*)(1 \le i \le n)$ from n users.

Output: An unbiased estimator F^* of the true frequency estimation on d-dimensional categorical data.

```
F* = {F_1^*, F_2^*, ..., F_d^*}, F_j^* denotes the frequency distribution on
  Attribute A_j;
for i = 1 to n do
    for j = 1 to d do
        for c ∈ C_j do
            if h_{i+j-1}(c) = y_i^* then
                F_jc^* = F_jc^* + 1;
for j = 1 to d do
    F_j^* = (1/n) · (F_j^* − nq)/(p − q);
return F*;
```

Operations on the curator side. On the curator side, curator computes the frequency distribution based on the reports from n users. In particular, for the report (h_i, y_i^*) from u_i, each dimension is traversed. For the jth dimension, hash function h_{i+j-1} with index $i+j-1$ is used to find the values in her domain C_j whose hash values are equal to y_i^*. If $h_{i+j-1}(k) = y_i^*$, F_{jk}^* increases by 1. The curator regards $\frac{F_j^* - nq}{p-q}$ as the unbiased frequency estimation, where $p = \frac{e^{\epsilon_l}}{e^{\epsilon_l}+g-d}$ and $q = \frac{1}{g}$. And the process of frequency computation is described by Algorithm 2.

Lemmas 3 and 4 discuss the privacy guarantee and variance of the published result provided by the component used on the user side in MHM.

Lemma 3. *Algorithm 1 satisfies ϵ_l-local differential privacy.*

Proof. For any output $(h_i, y_i^*) \in \mathcal{H} \times [g]$ and any two tuples $x_i, x_i' \in C_1 \times C_2 \times \cdots \times C_d$, we have

$$\frac{\Pr[\mathrm{MHM}_u(x_i) = (h_i, y_i^*)]}{\Pr[\mathrm{MHM}_u(x_i') = (h_i, y_i^*)]} = \frac{\frac{1}{|\mathcal{H}|} \cdot \Pr[y_i^* | \{h_i(x_{i1}), \ldots, h_{i+d-1}(x_{id})\}]}{\frac{1}{|\mathcal{H}|} \cdot \Pr[y_i^* | \{h_i(x_{i1}'), \ldots, h_{i+d-1}(x_{id}')\}]}$$

Based on Eq. (1), we observe that the ratio of maximum and minimum on the probability distribution of the perturbing mechanism is bounded by e^{ϵ_l}. So we have $\Pr[y_i^* | \{h_i(x_{i1}), \ldots, h_{i+d-1}(x_{id})\}] \le \exp(\epsilon) \Pr[y_i^* | \{h_i(x_{i1}'), \ldots, h_{i+d-1}(x_{id}')\}]$. Obviously, the above equation is also bounded by $\exp(\epsilon_l)$. □

Lemma 4. *The estimated frequency for the value m in the attribute A_j is unbiased and its variance is $\frac{q(1-q)+F_{jm}(p-q)(1-p-q)}{n(p-q)^2}$, where $p = \frac{e^{\epsilon_l}}{e^{\epsilon_l}+g-d}$ and $q = \frac{1}{g}$.*

Since this lemma can be proved in a similar way to that in CM (Wang et al., 2021a), we omit the proof. Similar with that in Ref. Wang et al. (2017), we approximate the variance of $Var_{\mathrm{MHM}}[F_{jm}^*]$ by Eq. (2)

Table 2
Comparisons on communication cost, computation cost from the user side and computation cost from the server side for FLAME, TCM and MHM. Δ denotes the communication cost of transmitting the encoding rule of attribute values.

Methods	FLAME	TCM	MHM
Comm.	$O\left(\log(dC_i)\right)$	$O(\log(ng))+\Delta$	$O(\log(ng))$
Comp.(user)	$O(1)$	$O(d)$	$O(d)$
Comp.(server)	$O(n)$	$O\left(n\sum_i C_i\right)$	$O\left(n\sum_i C_i\right)$

and then infer the optimal g value, i.e., g_{MHM}, as shown in Eq. (3).

$$Var_{\text{MHM}}[F^*_{jm}] \approx \frac{q(1-q)}{n(p-q)^2}. \tag{2}$$

$$g_{\text{MHM}} = d + \frac{de^{\epsilon_l}}{2} + \sqrt{2d^2e^{\epsilon_l} + \frac{d^2e^{2\epsilon_l}}{4} - 2de^{\epsilon_l}}. \tag{3}$$

Analysis of communication and computation costs. In MHM, each user needs to send two kinds of information, including the hash function index for the first dimension and the perturbed value. The latter can be encoded by $O(\log(g))$ bits. As for the hash function, it can be encoded using an index for the family $\mathcal{H}$ and takes $O(\log(n))$ bits. So the communication cost is $O(\log(gn))$ in total. The computation cost on the user side is $O(d)$, due to hashing d values. As for the curator side, the time cost is $O(n\sum|C_i|)$. Table 2 shows the theoretical communication cost, computation cost on the user(server) side for the methods including FLAME, TCM and MHM. It is observed that the compared FLAME outperforms the other two methods on both communication cost and computation cost. But its performance on the result utility is inferior, which is validated in the experiment part. Besides, MHM has a similar performance with TCM, except that TCM has a bigger communication cost due to the encoding operation for different attribute values.

4.2. Privacy amplification analysis for MHM

The skeleton of MHM under SDP is clear, but the hard nut to crack is analyzing the relationship between ϵ_l and (ϵ_s, δ_s). The key is to estimate the amplified privacy brought by the shuffling operation—the core component under SDP. Thus, in the following, we will discuss the privacy amplification of shuffling generated in MHM.

In MHM, the shuffler does the permutation on the outputs from n users, each of which is one pair consisting of a hash function for the first dimension and a perturbed value. Since the above pair is a basic unit involved in the permutation, we firstly show the probability mass function associated with the local randomizer on the basic unit's domain $\mathcal{H}\times[g]$, and then analyze some properties of this pmf to derive the amplified privacy.

Given the true data x_i of user u_i, let $r_{x_i}(h_i, y_i^*)$ be the probability that local randomizer outputs $(h_i, y_i^*) \in \mathcal{H}\times[g]$, whose value is shown in Eq. (4).

$$r_{x_i}(h_i, y_i^*) = \begin{cases} \frac{p}{|\mathcal{H}|}, & \text{if } y_i^* \in \mathbf{y}(x_i), \\ \frac{1-p\ell}{|\mathcal{H}|\cdot(g-\ell)}, & \text{if } y_i^* \notin \mathbf{y}(x_i). \end{cases} \tag{4}$$

where $\mathbf{y}(x_i) = \{h_i(x_{i1})\}\cup\{h_{i+1}(x_{i2})\},\ldots,\cup\{h_{i+d-1}(x_{id})\}$, and ℓ is the size of $\mathbf{y}(x_i)$.

Then $r_{x_i}(h_i, y_i^*)$ can be decomposed into

$$r_{x_i}(h_i, y_i^*) = (1-\gamma)v_{x_i}(h_i, y_i^*) + \gamma\omega(h_i, y_i^*)$$

where $v_{x_i}(h_i, y_i^*)$ shown in Eq. (5) is the distribution that depends on the true data x_i. And $\omega(h_i, y_i^*)$ is the blanket distribution which is a uniform distribution on $\mathcal{H}\times[g]$, i.e., $\omega(h_i, y_i^*) = \frac{1}{|\mathcal{H}|\cdot g}$. With probability $1-\gamma$, the output is dependent on the true data, where $\gamma = \frac{g}{de^{\epsilon_l}+g-d}$; and with probability γ, the output is random.

$$v_{x_i}(h_i, y_i^*) = \begin{cases} \frac{1}{|\mathcal{H}|\cdot d}, & y_i^* \in \mathbf{y}(x_i) \\ \frac{d-\ell}{|\mathcal{H}|\cdot d(g-\ell)}, & y_i^* \notin \mathbf{y}(x_i) \end{cases} \tag{5}$$

Recall that to derive the amplified privacy brought by shuffling, we need to use some information of *privacy amplification random variable* associated with $r_{x_i}(h_i, y_i^*)$ to compute the upper bound of the hockey-stick divergence of order e^{ϵ_s} between MHM(D) and MHM(D′). Theorem 1 discusses the settings of used information based on the pmf $r_{x_i}(h_i, y_i^*)$ of the local randomizer in MHM and its decompositions.

Theorem 1. *Given the local randomizer of MHM with privacy budget ϵ_l for any $\epsilon_s \geq 0$ and $x_i, x_i' \in C_1\times\cdots\times C_d$, the privacy amplification random variable $L_{\epsilon_s}^{x_i,x_i'}$ associated with the local randomizer satisfies:*

1. $\mathbb{E}L_{\epsilon_s}^{x_i,x_i'} = 1 - e^{\epsilon_s}$
2. $\gamma(1-e^{\epsilon_s}) - (1-\gamma)\frac{ge^{\epsilon_s}}{d} \leq L_{\epsilon_s}^{x_i,x_i'} \leq \gamma(1-e^{\epsilon_s}) + (1-\gamma)\frac{g}{d}$
3. $\mathbb{E}(L_{\epsilon_s}^{x_i,x_i'})^2 \leq (e^{2\epsilon_s}+1)\frac{p^2dg^2-2pdg+g}{g-d} - 2e^{\epsilon_s}\frac{g(1-pd)(g+gpd-2d)}{(g-d)^2}$

Proof. (1) Since the proof is similar to that in Balle et al. (2019) we omit them. (2) Next, we discuss the lower bound and upper bound of $L_{\epsilon_s}^{x_i,x_i'}$. Since $r_{x_i}(h_i, y_i^*)$ can be written as $(1-\gamma)v_{x_i}(h_i, y_i^*) + \gamma\omega(h_i, y_i^*)$, we have

$$\begin{aligned} L_{\epsilon_s}^{x_i,x_i'} &= \frac{r_{x_i}(h_i, y_i^*) - e^{\epsilon_s}r_{x_i'}(h_i, y_i^*)}{\omega(h_i, y_i^*)} \\ &= \gamma(1-e^{\epsilon_s}) + (1-\gamma)\cdot|\mathcal{H}|g\cdot\left[v_{x_i}(h_i, y_i^*) - e^{\epsilon_s}v_{x_i'}(h_i, y_i^*)\right] \end{aligned} \tag{6}$$

To derive the range of $L_{\epsilon_s}^{x_i,x_i'}$, we analyze the range of $v_{x_i}(h_i, y_i^*) - e^{\epsilon_s}v_{x_i'}(h_i, y_i^*)$ firstly. Specifically, as shown in Eq. (5), if $y_i^* \notin \mathbf{y}(x_i)$, the probability $v_{x_i}(h_i, y_i^*)$ is $\frac{d-\ell}{|\mathcal{H}|\cdot d(g-\ell)}$, which is changed with different ℓ. Let $f(\ell)$ be $\frac{d-\ell}{|\mathcal{H}|\cdot d(g-\ell)}$, where $\ell \in [1, d]$. Then the first-order derivative of $f(\ell)$ is as follows:

$$f'(\ell) = \frac{|\mathcal{H}|d(d-g)}{|\mathcal{H}|^2d^2(g-\ell)^2}.$$

Obviously, $f(\ell)$ is a monotonic decreasing function, since $d \leq g$. As a result, $f(\ell) \in [0, \frac{d-1}{|\mathcal{H}|d(g-1)}]$. On the other hand, when $y_i^* \in \mathbf{y}(x_i)$, $v_{x_i}(h_i, y_i^*)$ is $\frac{1}{|\mathcal{H}|\cdot d}$ bigger than $\frac{d-1}{|\mathcal{H}|d(g-1)}$. Then the range of $v_{x_i}(h_i, y_i)$ is $[0, \frac{1}{|\mathcal{H}|\cdot d}]$. Together with Eq. (6), we have

$$L_{\epsilon_s}^{x_i,x_i'} \in \left[\gamma(1-e^{\epsilon_s}) - (1-\gamma)\frac{ge^{\epsilon_s}}{d}, \gamma(1-e^{\epsilon_s}) + (1-\gamma)\frac{g}{d}\right]$$

(3) Finally, we discuss the upper bound for the expectation of $(L_{\epsilon_s}^{x_i,x_i'})^2$. Let $W \sim \omega(h_i, y_i^*)$. $\mathbb{E}(L_{\epsilon_s}^{x_i,x_i'})^2$ can be written as follows:

$$\begin{aligned} &\mathbb{E}\left[\left(\frac{r_{x_i}(W) - e^{\epsilon_s}r_{x_i'}(W)}{\omega(W)}\right)^2\right] \\ =&(e^{2\epsilon_s}+1)\mathbb{E}\left[\left(\frac{r_{x_i}(W)}{\omega(W)}\right)^2\right] - 2e^{\epsilon_s}\mathbb{E}\left[\frac{r_{x_i}(W)\cdot r_{x_i'}(W)}{(\omega(W))^2}\right] \end{aligned} \tag{7}$$

To derive the upper bound of $\mathbb{E}(L_{\epsilon_s}^{x_i,x_i'})^2$, we analyze the values of $\mathbb{E}\left[\left(\frac{r_{x_i}(W)}{\omega(W)}\right)^2\right]$ and $\mathbb{E}\left[\frac{r_{x_i}(W)\cdot r_{x_i'}(W)}{(\omega(W))^2}\right]$ respectively in the following. Note that we omit the details of the derivations for Eqs. (8) and (9) for brevity. Please refer to the supplementary material (Appendix A) for details.

Since $W \sim \omega(h_i, y_i^*)$, we have

$$\begin{aligned} \mathbb{E}\left[\left(\frac{r_{x_i}(W)}{\omega(W)}\right)^2\right] &= \sum_{W\in\mathcal{H}\times[g]}\left(\frac{r_{x_i}(W)}{\omega(W)}\right)^2\times\omega(W) \\ &\leq \frac{p^2dg^2 - 2pdg + g}{g-d} \end{aligned} \tag{8}$$

Next, we discuss the lower bound of $\mathbb{E}\left[\frac{r_{x_i}(W)\cdot r_{x_i'}(W)}{(\omega(W))^2}\right]$, which can be written as follows:

$$\mathbb{E}\left[\frac{r_{x_i}(W)\cdot r_{x_i'}(W)}{\omega(W)^2}\right] = \sum_{W\in\mathcal{H}\times[g]}\left(\frac{r_{x_i}(W)\cdot r_{x_i'}(W)}{\omega(W)^2}\right)\times\omega(W) = \frac{g(1-pd)(g+gpd-2d)}{(g-d)^2} \tag{9}$$

By taking Eqs. (8) and (9) into Eq. (7), we can derive the upper bound of $\mathbb{E}\left(L_{\epsilon_s}^{x_i,x_i'}\right)^2$ based on $\mathbb{E}\left[\left(\frac{r_{x_i}(W)}{\omega(W)}\right)^2\right]$ and $\mathbb{E}\left[\frac{r_{x_i}(W)\cdot r_{x_i'}(W)}{\omega(W)^2}\right]$. □

Taking the values provided by Theorem 1 into Lemma 1, we can derive the upper bound of the hockey-stick divergence of order e^{ϵ_s} between MHM(D) and MHM(D'). Following that, based on Lemma 2 which illustrates the relationship between the divergence and differential privacy, by making the upper bound smaller than δ_s, we can derive the amplified privacy measured by ϵ_s, which is expressed by ϵ_l and δ_s. And on the other hand, the privacy budget ϵ_l involved in the local randomizer can be written by ϵ_s and δ_s. Then given the requirement of privacy protection for the output of shuffler, (ϵ_s, δ_s), we can compute the reasonable noise scale (associated with ϵ_l) added by the underlying LDP component.

5. Shuffler-assisted personalized differential privacy

In practice, when the dataset comprises multiple users with different privacy expectations, MHM customized for the a uniform privacy protection cannot be directly applied. There are two intuitive methods to utilize MHM as a component. One is to enforce all to follow the most strictest privacy protection level $(\epsilon_{min}, \delta_{min})$ among users and then invoke MHM with such privacy budget, where $(\epsilon_{min}, \delta_{min})$ denotes the minimum privacy budget from all users. Obviously, in this way, the users with weaker privacy protection requirement, i.e., with privacy budget greater than $(\epsilon_{min}, \delta_{min})$, waste some budgets, and hence result in the loss of utility. The other way is to group the users with the same privacy budget and then invoke MHM parameterized by their uniform budget. Then multiple versions of estimation results from different groups can be combined to derive a more accurate one with the weight average technique (Li et al., 2012). Although this method can avoid the budget waste effectively, multiple invocations of MHM on disjoint users lead to fewer users involving in shuffling. That significantly limits privacy amplification since fewer outputs from local randomizer are used to hide the output of some user. The utility of final results is thereby poor.

To overcome the shortcomings of the above two intuitive methods, we propose the *overlapping group mechanism*, which organizes the MHM invocation with the goal of maximizing the group size, in order to enhance the privacy amplification, while simultaneously making full use of the privacy budgets of all users to eliminate the budget waste. The enhanced privacy amplification of course will improve the utility of published data.

Overlapping group mechanism (OGM). Instead of partitioning users into disjoint groups, the significant advantage of OGM is to form overlapped groups and hence increase group size on average. To achieve this goal, it partitions privacy budgets into splits and then accordingly groups users. Alg. 3 presents the high-level overview of OGM which works as a dispatcher of scheduling MHM invocations. By taking users' sensitive data and privacy budget specifications as the initial input, OGM chooses the minimum privacy budget (x_0, y_0) as the first split from users in the set $\mathcal{U}$. Note that $\mathcal{U}$ contains all users at the beginning. For the select budget split, OGM invokes MHM to collect data from users in $\mathcal{U}$ under (x_0, y_0)-DP. Then we can derive a version of frequency distribution estimation Fs^* associated with the amplified privacy budget (used in local randomizer) ϵ_l. Later, (Fs^*, ϵ_l) will be used as input of

Algorithm 3: Overlapping group mechanism

Input : A set of n users with privacy budget specifications $\{(\epsilon_i, \delta_i) : 1 \le i \le n\}$.
Output: An unbiased estimator F^* of the true frequency estimation on d-dimensional categorical data.

1 Let $\mathcal{U}$ be the set of all users;
2 $v \leftarrow 0$, Fset={};
3 **while** $\mathcal{U} \neq \emptyset$ **do**
4 Choose the minimum privacy budget (x_0, y_0) among users in $\mathcal{U}$;
5 //invoke MHM to collect data from users in $\mathcal{U}$ with budget (x_i, y_i);
6 $(Fs^*, \epsilon_l) \leftarrow$ MHM((x_i, y_i), $\mathcal{U}$);
7 Fset = Fset $\cup\{(Fs^*, \epsilon_l)\}$;
8 **for** *each user* $u_i \in \mathcal{U}$ **do**
9 $\epsilon_i = \epsilon_i - x_0$, $\delta_i = \delta_i - y_0$;
10 **if** $\epsilon_i = 0$ **then**
11 Remove u_i from $\mathcal{U}$;
12 $v \leftarrow v + 1$;
13 Derive the frequency distribution estimation F^* based on Fset with weight average technique;
14 **return** F^*;

the weight average technique (Li et al., 2012) to derive a more accurate version of published results. Note that in the process above, each user u_i has already consumed budget (x_0, y_0) to guarantee (x_0, y_0)-DP. Based on the sequential composition property (McSherry, 2009) of DP, in the following, only $(\epsilon_i - x_0, \delta_i - y_0)$ budget is remained for u_i, as shown in Line 9. Once the updated ϵ_i is equal to 0, we know the budget of u_i is used up and we cannot collect data from her any more. These users are thereby removed from $\mathcal{U}$, as shown in Line 11. Then OGM continuously selects the minimum budget (x_1, y_1) as the second split for remained users in $\mathcal{U}$, and invokes MHM again to get another version. Such procedures are repeated multiple rounds, until the budgets of all users are used up.

Obviously, a user in $\mathcal{U}$ can contributes values in multiple rounds of MHM invocations associated with different budget splits. MHM thereby can collect values from users as many as possible, in order to enhance the privacy amplification. At last, in Line 13 of Algorithm 3, OGM combines the multiple versions of frequency distribution estimations into a single one using the existing weight average technique (Li et al., 2012). The latter has the greatest accuracy and is regarded as the final result.

Fig. 2 shows an example for better understanding OGM. In particular, the user set consists of four users u_1, u_2, u_3 and u_4, each of which is with privacy budget specification $B_i = (\epsilon_i, \delta_i)$, where $B_1 < B_2 = B_3 < B_4$. Here we define $B_i < B_j$ when $\epsilon_i < \epsilon_j$ and $\delta_i < \delta_j$, $B_i = B_j$ when $\epsilon_i = \epsilon_j$ and $\delta_i = \delta_j$. In the first round, the strictest privacy protection level, i.e., the minimum privacy budget split $(\epsilon_{p1}, \delta_{p1})$ (equaling to B_1), is chosen as the privacy guarantee for MHM. Now data are collected from all users u_1-u_4. In this way, OGM provides excessive privacy protection for u_2-u_4. In the remaining rounds, there exist extra chances for OGM to collect data from them again, to further optimize the frequency distribution estimation. The remaining budgets for u_2, u_3 and u_4 are then decreased as $(\epsilon_{p2}, \delta_{p2})$, $(\epsilon_{p2}, \delta_{p2})$ and $(\epsilon_{p2} + \epsilon_{p3}, \delta_{p3} + \delta_{p3})$ respectively, where $\epsilon_{p2} = \epsilon_2 - \epsilon_1$, $\delta_{p2} = \delta_2 - \delta_1$, $\epsilon_{p2} + \epsilon_{p3} = \epsilon_4 - \epsilon_1$ and $\delta_{p2} + \delta_{p3} = \delta_4 - \delta_1$. Then the currently minimum budget $(\epsilon_{p2}, \delta_{p2})$, as another split, is used for MHM to collect data from $\{u_2, u_3, u_4\}$ in the second round. Similarly, following that, OGM invokes MHM with budget $(\epsilon_{p3}, \delta_{p3})$ to collect data from u_4. At last, OGM does weight average on these three versions to derive the final result.

6. Experiments

In this section, we evaluate our proposals MHM in Section 4 and OGM in Section 5 with the uniform and personalized privacy budget

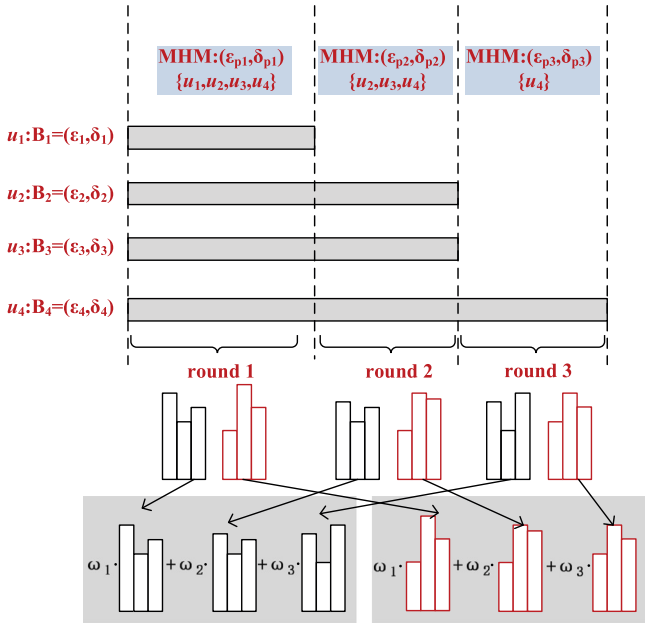

Fig. 2. An example to illustrate the overlapping group mechanism.

specifications respectively. All experiments were performed on an AMD Ryzen 3.6 GHz CPU with 16 GBytes of memory.

Competitors. We compare our proposals with the following protocols.

- Uniform privacy budget specifications: mainly including two state-of-the-art techniques for collecting multidimensional categorical data under SDP, FLAME (Liu et al., 2021) and TCM (Wang et al., 2021a). They are sample-based and transform-based methods, which give the bound of privacy amplification from shuffling based on the traditional DP definition (Balle et al., 2019). Please refer to Section 3.3 for details.
- Personalized privacy budget specifications: using the technique termed as non-overlapping group mechanism (NOGM) as the competitor. Since no existing solution can directly support this task, we take the following best-effort approach as the baseline. In particular, we firstly group the users by the budget specifications. Following that, MHM is invoked for each group of users with the same budget. Then weight average technique is used to combine different versions of frequency distribution estimations from different groups. Compared with our proposal OGM, this method considers the information of budgets firstly and splits users into non-overlapping groups based on budgets. So we term it as non-overlapping group mechanism.

Datasets. We use two public datasets extracted from the *Integrated Public Use Microdata Series*,[1] BR and MX. Both contain 100 000 census records but respectively from Brazil (with 10 attributes) and Mexico (with 14 attributes). Besides, to validate the effectiveness of MHM on the task for multidimensional range queries, we use extra two real datasets Adult,[2] and Loan[3] whose attributes are all with ordinal values. They have 30 thousands records with 5 attributes and 200 thousands records with 10 attributes respectively.

Utility Metric. We report result utility on three tasks under SDP including the classical frequency estimation, the k-way marginal query

[1] https://international.ipums.org.
[2] http://archive.ics.uci.edu/ml.
[3] https://www.kaggle.com/datasets/wordsforthewise/lending-club.

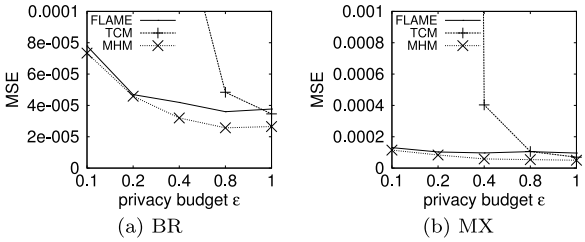

Fig. 3. Result accuracy for frequency estimation with real datasets.

as well as the range query, in terms of *mean square error* (MSE). The first is the focus of this paper. And the latter two are important research branches in DP and LDP, which have been investigated in a large number of works shown in Section 2. Usually, they adopt frequency distribution estimation as a module involved in their frameworks, where our proposal MHM can be utilized to optimize this module. So we also evaluate the performance of MHM on these two tasks. For the frequency distribution task, MSE is defined as $\frac{1}{d}\sum_{i=1}^{d}\frac{\sum_{j=1}^{|C_i|}(F_{ij}^*-F_{ij})^2}{|C_i|}$, where F_{ij}^* (F_{ij}) denotes the noisy (actual) frequency of the value j in the ith attribute. Similarly, in the k-way marginal query task, it is defined as $\frac{1}{C_d^k}\sum_{i=1}^{C_d^k}\frac{\sum_{j=1}^{|D_i|}(F_{ij}^*-F_{ij})^2}{|D_i|}$, where F_{ij}^* (F_{ij}) denotes the noisy (actual) frequency of the jth cell in the ith k-way marginal, and D_i denotes the domain of the ith k-way marginal. As for the range query, given a query set Q, MSE is defined as $\frac{\sum_{i=1}^{|Q|}(F_i^*-F_i)^2}{|Q|}$, where F_i^* (F_i) denotes the noisy (actual) result of query Q_i. A smaller MSE indicates that the noisy results returned by a technique are closer to the groundtruth. Besides, we also evaluate the efficiency of our proposals, in terms of the running time and communication cost.

6.1. Evaluation results for uniform privacy budget specifications

In this set of experiments, we evaluate the performance of our proposals MHM against FLAME and TCM on three analysis tasks, including the classical frequency estimation, k-way marginal queries and the multidimensional range queries.

6.1.1. Performance on frequency estimation

Fig. 3 plots the MSE results on frequency estimation with SDP as a function of the privacy budget ϵ with $\delta = \frac{1}{n}$. Overall, MHM consistently and significantly outperforms the state-of-the-art technique FLAME. When $\epsilon = 1$ on MX, our methods improve the MSE by about 0.5 times (on BR) or 1 times (on MX) compared with FLAME. Besides, we also observe that the gap between MHM and FLAME is proportional to ϵ. Also, MHM considerably outperforms TCM.

Discussion. The main reason for MHM beating FLAME is that the shuffler in FLAME adds a large number of dummy values, so that it can use the existing conclusion of privacy amplification based on bounded-size database. However, the added dummy values import extra errors. As for the gap between MHM and FLAME proportional to ϵ, it is because in FLAME, the error in the final estimation is brought by three operations, including adding dummy values, sampling and locally randomizing. By contrast, MHM just contains the latter one. On the other hand, the errors from the first two are independent of ϵ, while the latter is inversely proportional to ϵ. Together, when ϵ increases, the ratio of error brought by dummy values and sampling also increases, which leads to the gap between FLAME and MHM increased. Besides, the main reason for the superior performance of MHM to TCM is that the former takes a thorough analysis on her perturbation function, and fully utilizes the relationship between the divergence and differential privacy to explore the privacy amplification. Accordingly, they derive

Table 3
Comparisons of different methods on the metrics including true positive (TP for short), false positive (FP for short), true negative (TN for short) and false negative (FN for short).

Metrics	TP			FP			TN			FN		
Methods	FLAME	TCM	MHM	FLAME	TCM	MHM	FLAME	TCM	MHM	FLAME	TCM	MHM
$\epsilon = 0.1$	40	31	37	10	19	13	32	23	29	10	19	13
$\epsilon = 0.2$	39	37	42	11	13	8	31	29	34	11	13	8
$\epsilon = 0.4$	41	39	38	9	11	12	33	31	30	9	11	12
$\epsilon = 0.8$	41	38	41	9	12	9	33	30	33	9	12	9
$\epsilon = 1.0$	39	41	41	11	9	9	31	33	33	11	9	9

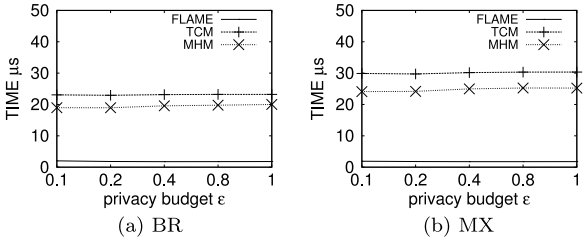

Fig. 4. Running time for frequency estimation with real datasets.

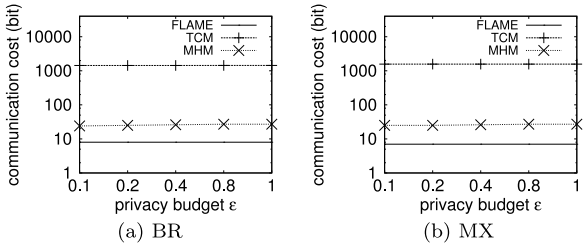

Fig. 5. Communication cost between one user and the shuffler for frequency estimation with real datasets.

much better bound of shuffling benefits, which leads to more accurate results.

Fig. 4 shows the running time of our proposals as well as the competitors on the user side. Note that the sampling-based FLAME significantly outperforms all the other methods, with a clear performance gap. We next investigate communication costs on the user side in Fig. 5, which measures the messages received and sent by the user. Observe that FLAME and MHM yield much lower communication costs than TCM. And FLAME outperforms MHM.

Discussion. The superior performance of FLAME on running time is that FLAME just perturbs the value from the single sampled dimension, while MHM needs to hash the value from each dimension. Processing values from fewer dimensions clearly leads to runtime reduction. Although FLAME shows the remarkable runtime superiority in our simulated experiments, it is not clear how to deploy its extra ability-empowered shuffler in real applications. In other words, there is doubt that it is reasonable to make the shuffler be able to count or add dummy values. Even though it can be reasonably implemented, empowering the shuffler more abilities might incur possibly potential attacks. Overall, in our opinion, the traditional SDP-based method TCM and MHM are still preferred solutions where the latter runs faster because of the elimination of encoding operations. Besides, the main reason for FLAME and MHM beating TCM on the communication cost metric is that for TCM, the curator needs to broadcast encoding rules to all users so that the latter can consistently encode values into distinct items, even though these values from different dimensions have the same expression. That clearly generates expensive communication costs. In our evaluation, we suppose that the domain of each attribute is not available on the user side and the encoding rule is to sequentially number domain values from 1 to $\sum_{i=1}^{d} |C_i|$. To make all users encode their own multidimensional data consistently, the curator sends the encoding rules to users. Then TCM at least incurs additional $\sum_{i=1}^{d} \left(|C_i| \cdot (8 + \log(\sum_{i=1}^{d} |C_i|)) \right)$ communication costs compared with MHM. Here we assume that each value from different dimension domains can be expressed by a char. In particular, FLAME outperforms MHM, because the latter uses $\log(n)$ bits to transmit the hash function information, instead of $\log(d)$ bits for the sampled dimension in the former.

Besides, we also show the performance of our proposal and the competitors on the metrics including true positive, false positive, true negative and false negative. Table 3 reports these metrics on the task of publishing top-50 frequent attribute values. It is observed that MHM performs similarly with FLAME, but is superior to TCM. Generally, a higher result accuracy on frequency distribution estimation leads to better performance on the above metrics computed on top-50 frequent attribute values. That is validated by the superiority of MHM to TCM. However, if the gap of result accuracy is smaller, the result accuracy cannot dominate the performance of publishing top-k frequent attribute values task, especially when the kth maximal frequency is much larger than the $(k+1)$th maximal frequency. As a result of that, although MHM performs better than FLAME on the result accuracy of frequency distribution estimation, they have a similar performance on the top-k frequent attribute value task.

6.1.2. Performance on k-way marginal queries

In this section, we study the performance of all solutions on k-way marginal queries. Note that Calm (Zhang et al., 2018), as one of the state-of-the-art related LDP-based techniques, publishes results on some views, each of which consists of carefully chosen attributes. Accordingly, any marginal query can be derived using the Maximum Entropy principle. Due to the effectiveness of Calm, we plug MHM into the its framework to do the task under SDP. Observe that in Clam, it is privacy-sensitive to only derive the frequency estimations on views. To make a comparison between MHM and the competitors, we keep its framework unchanged and just replace the frequency estimations component by MHM or competitors. In particular, each view, i.e. some attributes combination $\{S_1, \ldots, S_i\}$, is regarded as one dimension associated with $\prod_{j=1}^{i} |C_{S_j}|$ values, where S_j and C_{S_j} respectively indicate an attribute and its domain. In this way, the frequency distribution estimations on multiple views can be transformed into the ones on multiple dimensions, which is consistent with the problem that our paper focuses on.

Fig. 6 shows the MSE results on frequency estimation of the chosen views when answering 10 random 3-way marginals. Note that when testing k-way marginal queries and range queries, only MHM and FLAME are plotted, and TCM is left out, because the significant accuracy improvement of the former two have been validated in Fig. 3. From Fig. 6, we observe that the MSEs are also inversely proportional to budgets and MHM has a superior to FLAME, which agrees with the phenomenon shown in Section 6.1.1. Besides, Fig. 6 reveals the gap between MHM and FLAME for views is much larger than that for multiple dimensions in Fig. 3. In some settings, the MSE of FLAME

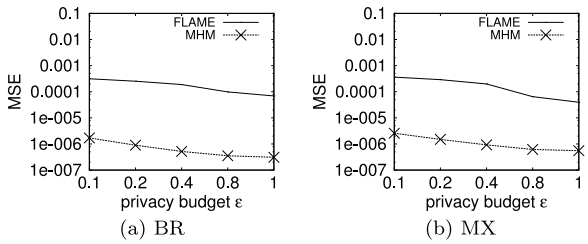

Fig. 6. Result accuracy of views for answering 3-way marginal queries with the framework of Calm.

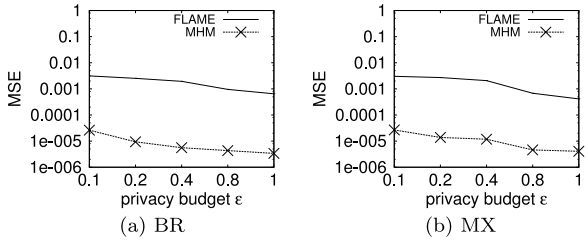

Fig. 7. Result accuracy of 3-way marginal queries with the framework of Calm.

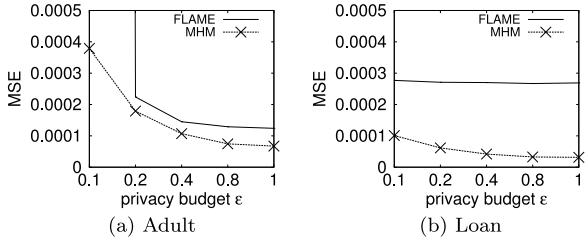

Fig. 8. Result accuracy of grids for answering range queries with the framework of HDG.

is about 100 times larger than that of MHM. Fig. 7 plots the MSE results on frequency estimations on 3-way marginal queries which are answered by the estimations on views. Undoubtedly, Fig. 7 shows the same phenomenon as that in Fig. 6, except that the noise scale is bigger than that for views.

Discussion. The main reason for the inferior performance of FLAME in Figs. 6 and 7 is that the amplified privacy budget ϵ_l for this view provided by FLAME is equal to $\log\left(\frac{(N_p-1)\epsilon_{ck}^2}{14\log(2/\delta_{ck})}-k+1\right)$, where k denotes the domain size. On the other hand, the domain size of a view is the product of the ones of covering attributes (dimensions), which is much bigger and sometimes up to 4000 on these two datasets, leading to a smaller amplified budget for the view. Then, more noise is added by the local randomizer and the result utility becomes poor. Besides, as for the bigger noise scale for 3-way marginal query than that for views, that is due to the fact that the result of 3-way marginal query is derived from the frequency estimations on views by the Maximum Entropy principle with the framework of Calm. So the result utility on views dominates the one on marginal queries. Besides, this Maximum Entropy principle is an approximate way to derive the results of k-way marginal queries, which also imports error. That is why the noise scale in Fig. 7 is bigger.

6.1.3. Performance on multidimensional range queries

We finally evaluate the performance of our proposal when answering multidimensional range queries. HDG (Yang et al., 2020) is

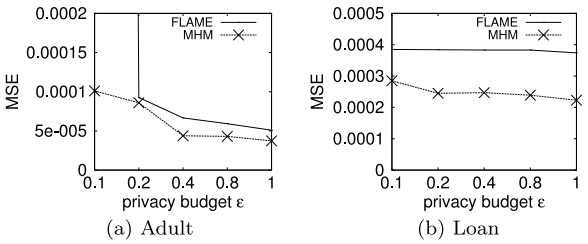

Fig. 9. Result accuracy of 3-dimensional range queries with the framework of HDG.

the up-to-date LDP-compliant technique for multidimensional range queries. Similar with Calm (Zhang et al., 2018) mentioned above, it also derives the answers based on published frequency distribution estimation on LDP-compliant sketches. Here, the sketch consists of both one-dimensional and two-dimensional grids. In particular, given two parameters g_1 and g_2, HDG uniformly partitions the one-dimensional domains of each attribute into g_1 grids and two-dimensional domains of all attribute pairs into g_2 grids. And then it invokes the LDP component to collect data from users to derive the frequency of each grid. Like Calm, HDG is also privacy-sensitive. We then invoke FLAME or MHM to replace its LDP component, in order to make HDG work under SDP. Specifically, we regard a one-dimensional grid as an attribute with g_1 values and two-dimensional grid as an attribute with g_2 values. Now the problem of frequency estimations on grids is transformed into the one on multidimensional data.

Figs. 8 and 9 presents the MSE results on frequency estimations on $d + C_d^2$ grids as well as 10 random 3-dimensional range queries respectively. MHM consistently outperforms FLAME. The MSE results are both inversely proportional to ϵs and the reason has been discussed in Section 6.1.1. We omit it for brevity. Another interesting observation is that the MSEs of multidimensional range queries are sometimes smaller than those on grids, which is different from the phenomenon shown in Section 6.1.2. That is because HDG uses frequency estimations on each attribute pair, rather than the ones on grids, to answer range queries. The former is derived by postprocessing the estimations on grids, in which to some extent the operation can be regarded as doing consistency among the frequency estimations on grids. As we all know, consistency-enforced estimation has a positive effect on the result utility.

6.2. Evaluation results for personalized privacy budget specifications

Finally, we verify the effectiveness of OGM. To simulate the non-uniform privacy specifications, we manually divide users into three groups in a random manner: conservative users with a low privacy budget (ϵ_c, δ_c) as $(0.1, \frac{1}{10n})$; moderate users with a medium privacy budget (ϵ_m, δ_m) as $(0.5, \frac{1}{n})$; and liberal users with a high privacy budget (ϵ_l, δ_l) as $(1, \frac{10}{n})$.

Fig. 10 plots the MSE as a function of the ratio for the numbers of conservative, moderate and liberal users. Overall, our proposal OGM consistently and significantly outperforms the best-effort approach NOGM. Notably, when the ratio is 2:4:4 on the MX dataset, the MSE of NOGM is one times larger than the one of OGM. Moreover, it is observed that with the increasing of the number of moderate or liberal users, both two methods perform better. That agrees with the fact that an average lower privacy concern means less noise, and then can generate higher utility results.

Discussion. The main reason for OGM's superior performance is that in MHM the scale of privacy amplification is proportional to the number of users' outputs involved in the shuffling operation, since more participants can hide one of them more easily. OGM adequately utilizes

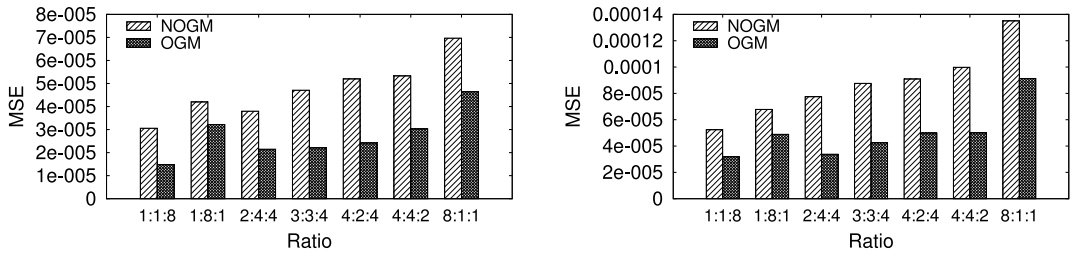

Fig. 10. Result accuracy for frequency estimations under personalized privacy budget specifications with real datasets.

the knowledge and always attempts to maximize the number of users involved in each invocation of MHM. Consequently, OGM usually has a bigger amplified privacy budget, leading to less noise added by the local randomizer. That contributes to improving the result utility of frequency estimations.

7. Conclusion

This work investigates the problem of collecting multidimensional categorical users' personal data under the SDP model. We propose Multiple Hash Mechanism to deal with this problem, as well as a thorough shuffling benefit analysis method for this mechanism. Extensive experiments demonstrate the effectiveness of our proposals. In the next step, we plan to extend our proposals to publish mean estimations on the multidimensional numerical data, which can be applied for publishing the gradients involved in training the machine learning model.

CRediT authorship contribution statement

Ning Wang: Methodology. **Jian Zhuang:** Validation. **Zhigang Wang:** Writing – original draft. **Zhiqiang Wei:** Conceptualization. **Yu Gu:** Investigation. **Peng Tang:** Writing – review & editing. **Ge Yu:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the Key R&D Program of Shandong Province, China (No. 2023CXPT020), the National Natural Science Foundation of China (Grant Nos. 61902365 and 61902366), and Open Project Program from Key Lab of Cryptologic Technology and Information Security, Ministry of Education, Shandong University, China.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cose.2024.104301.

Data availability

Data will be made available on request.

References

Aggarwal, C.C., 2005. On k-anonymity and the curse of dimensionality. In: Böhm, K., Jensen, C.S., Haas, L.M., Kersten, M.L., Larson, P., Ooi, B.C. (Eds.), VLDB. pp. 901–909.

Agrawal, R., Srikant, R., 1994. Fast algorithms for mining association rules in large databases. In: VLDB. pp. 487–499.

Balle, B., Barthe, G., Gaboardi, M., 2018. Privacy amplification by subsampling: Tight analyses via couplings and divergences. In: NeurIPS. pp. 6280–6290.

Balle, B., Bell, J., Gascón, A., Nissim, K., The privacy blanket of the shuffle model, in: CRYPTO, 638–667.

Barthe, G., Olmedo, F., 2013. Beyond differential privacy: Composition theorems and relational logic for f-divergences between probabilistic programs. In: ICALP. Vol. 7966, pp. 49–60.

Bittau, A., Erlingsson, Ú., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnés, J., Seefeld, B., 2017. Prochlo: Strong privacy for analytics in the crowd. In: Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China,. Vol. 28–31, pp. 441–459.

Caruccio, L., Desiato, D., Polese, G., Tortora, G., Zannone, N., 2022. A decision-support framework for data anonymization with application to machine learning processes. Inform. Sci. 613, 1–32.

Chen, R., Li, H., Qin, A.K., Kasiviswanathan, S.P., Jin, H., 2016. Private spatial data aggregation in the local setting. In: ICDE. pp. 289–300.

Cheu, A., Smith, A.D., Ullman, J.R., Zeber, D., Zhilyaev, M., 2019. Distributed differential privacy via shuffling. In: Ishai, Y., Rijmen, V. (Eds.), EUROCRYPT. pp. 375–403.

Du, L., Zhang, Z., Bai, S., Liu, C., Ji, S., Cheng, P., Chen, J., 2021. AHEAD: adaptive hierarchical decomposition for range query under local differential privacy. In: Kim, Y., Kim, J., Vigna, G., Shi, E. (Eds.), CCS. pp. 1266–1288.

Dwork, C., Kenthapadi, K., McSherry, F., Mironov, I., Naor, M., 2006a. Our data, ourselves: Privacy via distributed noise generation. In: Advances in Cryptology-EUROCRYPT 2006. Springer, pp. 486–503.

Dwork, C., McSherry, F., Nissim, K., Smith, A., 2006b. Calibrating noise to sensitivity in private data analysis. In: TCC. pp. 265–284.

Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., Thakurta, A., 2019. Amplification by shuffling: From local to central differential privacy via anonymity. In: Chan, T.M. (Ed.), SODA. SIAM, pp. 2468–2479.

Erlingsson, Ú., Pihur, V., Korolova, A., 2014. Rappor:randomized aggregatable privacy-preserving ordinal response. In: CCS. pp. 1054–1067.

Fanti, G.C., Pihur, V., Erlingsson, Ú., 2016. Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries. PoPETs 2016 (3), 41–61.

Feldman, V., McMillan, A., Talwar, K., 2021. Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In: FOCS. pp. 954–964.

Garg, S., Torra, V., 2024. Privacy in manifolds: Combining k-anonymity with differential privacy on fréchet means. Comput. Secur. 144, 103983.

Ghazi, B., Manurangsi, P., Pagh, R., Velingker, A., 2020. Private aggregation from fewer anonymous messages. In: EUROCRYPT. Vol. 2020, pp. 798–827.

Jorgensen, Z., Yu, T., Cormode, G., 2015. Conservative or liberal? personalized differential privacy. In: Gehrke, J., Lehner, W., Shim, K., Cha, S.K., Lohman, G.M. (Eds.), ICDE. pp. 1023–1034.

Li, X., Liu, W., Chen, Z., Huang, K., Qin, Z., Zhang, L., Ren, K., DUMP: A dummy-point-based framework for histogram estimation in shuffle model, CoRR abs/2009.13738.

Li, N., Qardaji, W., Su, D., Cao, J., 2012. Privbasis: Frequent itemset mining with differential privacy. PVLDB 5 (11), 1340–1351.

Li, H., Xiong, L., Ji, Z., Jiang, X., 2017. Partitioning-based mechanisms under personalized differential privacy. In: PAKDD. pp. 615–627.

Liu, R., Cao, Y., Chen, H., Guo, R., Yoshikawa, M., 2021. FLAME: differentially private federated learning in the shuffle model. In: AAAI. pp. 8688–8696.

Liu, X., Liu, Q., Wang, J., Sun, H., 2024. Multidimensional epidemiological survey data aggregation scheme based on personalized local differential privacy. Symmetry 16 (3), 294.
Machanavajjhala, A., Gehrke, J., Kifer, D., Venkitasubramaniam, M., 2006. L-diversity: Privacy beyond k-anonymity. In: ICDE. p. 24.
McSherry, F., 2009. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: SIGMOD. pp. 19–30.
Nie, Y., Yang, W., Huang, L., Xie, X., Zhao, Z., Wang, S., 2019. A utility-optimized framework for personalized private histogram estimation. IEEE Trans. Knowl. Data Eng. 31 (4), 655–669.
Niu, B., Chen, Y., Wang, B., Wang, Z., Li, F., Cao, J., 2021. Adapdp: Adaptive personalized differential privacy. In: INFOCOM. pp. 1–10.
Qin, Z., Yang, Y., Yu, T., Khalil, I., Xiao, X., Ren, K., 2016. Heavy hitter estimation over set-valued data with local differential privacy. In: CCS. pp. 192–203.
Quinlan, J.R., 1986. Induction of decision trees. Mach. Learn. 1 (1), 81–106.
Ren, X., Yu, C., Yu, W., Yang, S., Yang, X., McCann, J.A., Yu, P.S., 2018. Lopub: High-dimensional crowdsourced data publication with local differential privacy. IEEE Trans. Inf. Forensics Secur. 13 (9), 2151–2166.
Scott, M., Cormode, G., Maple, C., 2022. Aggregation and transformation of vector-valued messages in the shuffle model of differential privacy. IEEE Trans. Inf. Forensics Secur. 17, 612–627.
Wang, T., Blocki, J., Li, N., Jha, S., 2017. Locally differentially private protocols for frequency estimation. In: USENIX Security. pp. 729–745.
Wang, T., Ding, B., Zhou, J., Hong, C., Huang, Z., Li, N., Jha, S., 2019a. Answering multi-dimensional analytical queries under local differential privacy. In: SIGMOD. pp. 159–176.
Wang, S., Li, J., Qian, Y., Du, J., Lin, W., Yang, W., 2021a. Hiding numerical vectors in local private and shuffled messages. In: IJCAI. pp. 3706–3712.
Wang, N., Xiao, X., Yang, Y., Zhao, J., Hui, S.C., Shin, H., Shin, J., Yu, G., 2019b. Collecting and analyzing multidimensional data with local differential privacy. In: ICDE. pp. 638–649.
Wang, T., Xu, M., Ding, B., Zhou, J., Hong, C., Huang, Z., Li, N., Jha, S., 2020. Improving utility and security of the shuffler-based differential privacy. Proc. VLDB Endow. 13 (13), 3545–3558.
Wang, T., Yang, X., Ren, X., Yu, W., Yang, S., 2022. Locally private high-dimensional crowdsourced data release based on copula functions. IEEE Trans. Serv. Comput. 15 (2), 778–792.
Wang, T., Zhao, J., Hu, Z., Yang, X., Ren, X., Lam, K., 2021b. Local differential privacy for data collection and analysis. Neurocomputing 426, 114–133.
Warner, S.L., 1965. Randomised response: a survey technique for eliminating evasive answer bias. J. Amer. Statist. Assoc. 60 (309), 63–69.
Yang, J., Wang, T., Li, N., Cheng, X., Su, S., 2020. Answering multi-dimensional range queries under local differential privacy. Proc. VLDB Endow. 14 (3), 378–390.
Zhang, Z., Wang, T., Li, N., He, S., Chen, J., 2018. CALM: consistent adaptive local marginal for marginal release under local differential privacy. In: CCS. pp. 212–229.

Ning Wang received the Ph.D. degree in computer software and theory from Northeastern University, China, in 2017. She is currently an associate professor in the Cyberspace Institute of Advanced Technology, Guangzhou University in China. Her research interests include data privacy and machine learning.

Jian Zhuang received the BE degree in Computer Science and Technology from Wuhan University of Technology, China, in 2020. He is working towards the PHD degree in the College of Information Science and Engineering, Ocean University of China. His research interests include data privacy.

Zhigang Wang received the Ph.D. degree in computer software and theory from Northeastern University, China, in 2018. He is currently an associate professor in the Cyberspace Institute of Advanced Technology, Guangzhou University in China. His research interests include cloud computing, distributed graph processing and machine learning. He is a member of the China Computer Federation (CCF). He received the CCF Outstanding Doctoral Dissertation Award in 2018.

Zhiqiang Wei received the Ph.D. degree from Tsinghua University, China, in 2001. He is currently a professor with the Ocean University of China. He is also the director of High Performance Computing Center at the Pilot National Laboratory for Marine Science and Technology(Qingdao). His current research interests are in the fields of intelligent information processing,social media and big data analytics. He is a member of IEEE and CCF.

Yu Gu received the Ph.D. degree in computer software and theory from Northeastern University, China, in 2010. Currently, he is a professor and the Ph.D. supervisor at Northeastern University, China. His current research interests include big data analysis, spatial data management and graph data management. He is a senior member of the China Computer Federation (CCF).

Peng Tang received the Ph.D. degree in 2019 from the Beijing University of Posts and Telecommunications, China. He is currently a Assistant Professor at Shandong University. His research interests include data privacy and databases.

Ge Yu received the Ph.D. degree in computer science from Kyushu University of Japan, in 1996. He is currently a professor and the Ph.D. supervisor at Northeastern University of China. His research interests include distributed and parallel database, OLAP and data warehousing, data integration, graph data management, etc. He is a member of ACM, a senior member of IEEE, and a Fellow of the China Computer Federation(CCF).