Fast top-k frequent itemset mining under Local Differential Privacy*

*Note: Sub-titles are not captured in Xplore and should not be used

1st Wang JiaLi
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

***Abstract*—This is the abstract.**

***Index Terms*—This is the keywords**

I. Introduction

Differential privacy (DP) [7] is the state-of-the-art approach that is used to protect individual privacy in the process of data collection, which has been named one of the world's top 10 breakthrough technologies in 2020 by the MIT technology review. It is a means in cryptography that aims to provide a way to maximize the accuracy of data queries when querying from statistical databases while minimizing the chances of identifying their records. Meanwhile, as a mathematical technique, it can add noise to the data while quantifying the extent of the increase in privacy, thus making the process of adding "noise" more rigorous.

Due to its unique advantages, DP has been widely studied by the academia and industry. For example, Google, Microsoft, apple and other companies use this technology to protect users' privacy, and at the same time, mobile phones aggregate data, so as to improve service quality. And the U.S. government is to complete a census of 330 million U.S. residents by 2020, keeping their identities secret, in what would be the largest application of DP ever.

There are two types of differential privacy - Centralized differential privacy (CDP) and Local differential privacy (LDP). Compared with CDP, the LDP does not require the assumptions of a trusted third party and provides stronger privacy guarantees. DP's research has involved many aspects, in recent years, the work in mining frequent itemsets has attracted the attention, which is one of the most important techniques because of its ability to locate the repeating relationships between different items in a data set and plays an essential role in mining association rules [9]. Formally, let $\mathcal{X} = \{x_1, x_2, ..., x_d\}$ be the global domain of items with size is d and $\mathcal{T} = \langle T_1, T_2, ..., T_n \rangle$ denote a transaction database for n users, where $T_i(i \in [1...n])$ denotes a transaction that is a subset of $\mathcal{X}$. For example, a sample of transational data is shown in Table I. The support of an itemset X, where $X \subseteq \mathcal{X}$ is a set of items, is the number of transactions containing X in $\mathcal{T}$. Then, given a minimum support threshold δ, the problem of finding the complete set of frequent itemsets that supports no less than δ is called the frequent itemset mining (FIM) problem.

TABLE I
Sample of transactional data.

TID	List of items
T01	a, f, c, g, p
T02	a, b, c, f, l, o
T03	b, f, h, o
T04	b, c, p
T05	f, a, c, l, p, n

A lot of work [3]–[6] has been done to solve DM problems in CDP. However, since the analyst holds the user's raw data in CDP setting, its main job is to add noise to the results to satisfy the DP definition.

In this paper, we consider the $top - k$ FIM problem in transaction databases under LDP. Although LDP is similar to CDP, the LDP has no reliance on third party. The data analyst wants to find k itemsets with highest support while while users are sensitive and unwilling to answer their real infomation. The main challenge is that the analyst does not hold the user's original sensitive information, which makes it quite difficult to mine useful information with sanitized data. Qin et al. [1] point out that if utlize directly existing FIM algorithm (e.g. Apriori [9], [10], FP-growth [8], Eclat [13]) would result in accumulation of dramatic noise because of multi-iteration between users and analyst.

Specifically for FIM in the local setting, Qin et al. [1] leave it as a future work but there is no clear solution. Wang et al. [2] solves the $top - k$ frequent itemset mining (FIM) task for the first time with **padding-and-sampling-based frequency oracle** (PSFO). In [2], the Set-Value Item Mining (SVIM) protocol is proposed to handles set values under the LDP setting, with the purpose of finding the k most frequent items and their frequencies. To mine frequent itemsets , a core technique is "Guessing Frequency (GF)". That is, the analyst first calculated the frequency of a given itemset X for all

Identify applicable funding agency here. If none, delete this.

candidate itemsets by

$$\mathbb{G}(X)=\prod_{x\in X}\mu(x),\mu(x)=\frac{0.9\times\tilde{\theta}(x)}{\max_{x\in S'}\tilde{\theta}(x)} \quad (1)$$

where $\mathbb{G}(X)$ represents the guessing frequency of itemset X, S' and $\tilde{\theta}(x)$ are denoted separately the $top-k$ frequent items set and the frequency of a given item x. Then $2k$ itemsets with highest guessing frequencies are selected to construct candidate set IS. Finally, it utilizes SVIM protocol again with the domain IS to mine $top-k$ itemsets. We observe that, the size of candidate set to construct IS increase significantly with k. As a result, it is computationally expensive when k is large (e.g., $k=100$).

Inspiringly, to reduce the expensive computational cost caused by guessing frequencies of exponentially possible itemsets, we introduce the frequent-pattern tree (FP-tree) [8] structure to store compactly sensitive transactions, and then FP-growth [8] is awakened to mine frequent itemsets over the tree. Summarily, we propose Frequent-Pattern-based miner (fpminer) protocol for discovering $top-k$ most frequent itemsets and their frequencies under LDP. It has five steps. First, the SVIM protocol is used to estimate the k most frequent items and their frequencies. Second, users report the number of frequent items they have; the analyst estimates the distribution user reported and figure out the right M as the maximum iteration of the tree. Third, users interact with the analyst to build effectively the FP-tree [8]. Fourth, the analyst optimizes and mines the FP-tree. Fifth, the analyst publishes $top-k$ itemsets.

Experimental results show that minefp outperforms SVSM in that it identifies quickly frequent itemsets as well as estimates the frequencies more accurately. Notably, when k is large, it provides significantly lower computation overhead as well as similar accuracy than existing SVSM protocol .

To summarize, the main contributions of this paper are:

- We study the application of FP-growth algorithm and design the FP-tree-based-mine (minefp) protocol to find frequent itemsets as well as their frequencies in the LDP setting. Experimental results on real-world datasets show the significant improvement over previous techniques.
- We investigate GF to construct candidate set and point out that it is beneficial to build hierarchically FP-tree.

Roadmap.

II. Preliminaries

A. Local Differential Privacy (LDP)

In the local setting, there is no trusted third party and an aggregator wants to gather information from users, where each user possesses an input t. The privacy of the data contributor is protected by perturbing her/his original data at the data contributor's side; thus, the agregator cannot access the original data, but is still able to obtain population statistics.

Formally, let $\mathcal{T}$ denote the whole transaction databases. ϵ-local differential privacy (or ϵ-LDP) is defined on an algorithm $\mathcal{A}$ and a privacy budget $\epsilon\geq 0$ as follows.

Definition 1 *($\epsilon-LDP$). A randomized algorithm $\mathcal{A}$ satisfies ϵ-local differential privacy (ϵ-LDP), if and only if for (1) all pairs of input $t_i,t_j\in\mathcal{T}$, and (2) any possible output $\mathcal{O}$ of $\mathcal{A}$, we have:*

$$\frac{\mathbf{Pr}[\mathcal{A}(t_i)=\mathcal{O}]}{\mathbf{Pr}[\mathcal{A}(t_j)=\mathcal{O}]}\leq e^{\epsilon}$$

$Sequential\ composability$ [12] and $post-processing$ [14] are vitally important properties of differential privacy. The former allows each user to divide privacy budget into multiple portions and use each portion to execute independent LDP protocols on the same input while the sequential executions provide $\sum\epsilon_i$-LDP; the latter guarantees that any processing of the noisy data do not disclose the privacy.

Theorem II.1 ($sequential\ composability$). *Given m randomized algorithms $\mathcal{A}_i(1\leq i\leq m)$, each of which satisfies ϵ_i-local differential privacy. Then the sequence of $\mathcal{A}_i$ collectively provides $(\sum_{i=1}^{m}\epsilon_i)$-local differential privacy.*

Theorem II.2 ($post-processing$). *For any method ϕ which works on the output of a ϵ-LDP algorithm $\mathcal{A}$ without accessing the raw data, the procedure $\phi\big(\mathcal{A}(\cdot)\big)$ remains ϵ-LDP.*

B. Frequency Oracle in the LDP setting

A frequency oracle (FO) protocol under LDP enables aggregator to estimate the frequency of any given value $t\in\mathcal{X}$ from all sanitized data recived from the users. A fundamental FO protocol is Randomized Response (RR) [15], which is a traditional technique for estimating unbiasedly a population proprotion. It is the building block of many sophisticated LDP protocols, such as RAPPOR [16], GRR and OLH [11]. Suppose the respondents were asked to answer a sensitive Boolean question (e.g. have you ever cheated on your partner?) in a survey, then RR makes provisions for each person to be interviewed. That is, each respondent gives the raw answer with probability p and gives the opposite answer with probabilit $q=1-p$. Specially, to satisfy ϵ-LDP, the probability p is set to $\frac{e^{\epsilon}}{1+e^{\epsilon}}$.

Then, the agregator calculates the estimated percentage of "Yes" (denote as $\tilde{\theta}$) as example from all sanitized answers, which is unbiased, as follows:

$$\tilde{\theta}_{RR}(Yes)=\frac{\mathcal{C}(answer=Yes)-nq}{p-q}$$

where n is the total number of respondents and $\mathcal{C}(answer=Yes)$ denotes the number of occurrences respondents answered "Yes". Accordingly, the variance of it is

$$Var\big[\tilde{\theta}_{RR}(Yes)\big]=$$

However, the RR protocol only applies to binary Boolean problems, which greatlt limits its application. Therefore, in [11], two effective protocols, Generalized Random Response (GRR) and Optimized Local Hash (OLH), are proposed for the purpose of solving problems with large domain size $d=|\mathcal{X}|$.

Generalized Random Response (GRR) [11]: GRR extends the RR protocol in case of $d \neq 2$ by setting probability $p = \frac{e^\epsilon}{e^\epsilon + d - 1}$ to give the raw answer $y = t$ and probability $q = \frac{1-p}{d-1}$ (i.e. $q = \frac{1}{e^\epsilon + d - 1}$) to give the perturbed answer $y \neq t$. Specially, it has been shown that RR is the special case while $d = 2$. The shortage of GRR is that its estimated variance is linear with d, which makes poor perform when d is large:

$$Var\big[\tilde{\theta}_{GRR}(t)\big] = n \cdot \frac{d - 2 + e^\epsilon}{(e^\epsilon - 1)^2} \tag{2}$$

Optimized Local Hashing (OLH) [11]: In order to deal with a large domain size d as well as reduce the communication cost, OLH protocol applys a hash function to map each input value into a value in $[g]$, where $g \geq 2$ and $g \ll d$. Then randomized response is used to the hashed value in the smaller domain. In [11], the optimal choice of the parameter g is $\lceil e^\epsilon + 1 \rceil$ which meets the minimal variance.

Let H is randomly chosen from a family of hash functions that outputs a value in $[g]$ and $x = H(t)$. The perturbing protocol in OLH is $Perturb_{OLH}\big(\langle H, x\rangle\big) = \langle H, y\rangle$, where

$$\forall_{i \in [g]} \mathbf{Pr}[y = i] = \begin{cases} p = \frac{e^\epsilon}{e^\epsilon + g - 1}, & \text{if } x = i \\ q = \frac{1}{e^\epsilon + g - 1}, & \text{if } x \neq i \end{cases}$$

Accordingly, the aggregator first calculates the number of perturbed values that "supports" that the input is t (denote as $\mathcal{C}(t)$), then transforms $\mathcal{C}(t)$ to its unbiased estimation

$$\tilde{\theta}_{OLH}(t) := \frac{\mathcal{C}(t) - n/g}{p - 1/g} \tag{3}$$

The variance of this estimation is

$$Var\big[\tilde{\theta}_{OLH}(t)\big] = n \cdot \frac{4e^\epsilon}{(e^\epsilon - 1)^2} \tag{4}$$

Intuitively, OLH has a variance that does not depend on d. However, the bigger d is, the more hash collisions there are, resulting in large errors. In [11], it suggests that when domain size $d < 3e^\epsilon + 2$, GRR is the best among all approaches; but for large d, OLH meets high accuracy as well as low communication cost

$- - - - - - - - - - - - -$ In [11], the two best performing FO protocols are Generalized Random Response (GRR) and Optimized Local Hash (OLH). The former extends the randomized response (RR) technique [15], which is an old technique developed for the interviewees in a survey to give random answer to a sensitive boolean question so that they can achieve plausible deniability; the latter deals with a large domain size $|I|$ by hashing a large domain size $|I|$ to a smaller size g and applying RR to the hashed value. It was found that GRR offers the best accuracy than OLH when $|I| < 3e^\epsilon + 2$, where $|I|$ is the size of the domain of items under consideration. Therefore, in [2], an adaptive FO protocol can be proposed from the known I as well as ϵ.

$- - - - - - - - - - - - -$

C. FP-growth algorithm

Frequent pattern growth (FP-growth) [8] is an algorithm that mines the complete set of frequent patterns without a costly candidate generation process, which based on the frequent pattern tree (FP-tree) structure that is an extended prefix-tree structure for storing compressed, crucial information about frequent patterns. The FP-Tree is further divided into a set of Conditional FP-Trees for each frequent item so that they can be mined separately. An example of the FP-Tree that represents the frequent items is shown in Fig. 1, where the minimum support threshold is set to 3.

The FP-growth algorithm solves the problem of identifying long frequent itemsets by searching through smaller conditional FP-tree repeatedly. The conditional pattern base is a "sub-database" which consists of every prefix path in the FP-Tree that co-occurs with every frequent length-1 itemset. It is used to construct the conditional FP-tree and generate all the frequent patterns related to the length-1 items. In this way, the cost of searching for the frequent patterns is substantially reduced.

Definition 2 *(length $-\ \alpha$ itemset).* *is this necessary?*

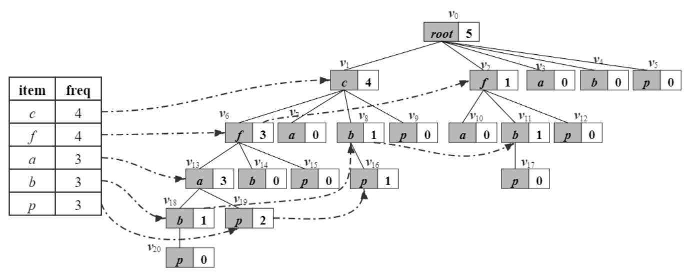

Fig. 1. Frequrnt pattern tree(FP-tree).

III. Problem Overview

A. Problem Definition

In this paper, we study the FIM problem while fully considering users' personal privacy. Or rather, we look at privacy preserving frequent itemset mining under the LDP setting. In general, let $\mathcal{X} = \{x_1, x_2, ..., x_d\}$ denote the whole domain of items, X is a length-$|X|$ itemset where $X \subseteq \mathcal{X}$ is a subset of $\mathcal{X}$ and $|X|$ denote the cardinality of it.

There are n users, and each user has a sensitive transaction, which is a subset of $\mathcal{X}$. Formally, the transaction of i-th user is $T_i (i \in [1, n])$ and $\mathcal{T} = \langle T_1, T_2, ..., T_n\rangle$ denote the whole transaction database. An untrusted data analyst (or aggregator) is going to mining $top - k$ most tfrequent itemsets and their frquencies among all users under ϵ-LDP. More specifically, we aim to mine k length-α itemset with highest frequencies, where $\alpha \geq 2$.

Table II lists the notations used in this paper

B. Existing Approaches to FIM under LDP

Since Qin et al. [1] first introduce set-valued data to differential privacy in the local setting for the purpose of

TABLE II
NOTATIONS.

Symbol	Description
n	the number of users
X	the itemset
$\|X\|$	the cardinality of itemset X
$\mathcal{T}$	the transaction database
T_i	the transaction of user i
$\mathcal{X}$	the set of items
d	the number of items, $d = \|\mathcal{X}\|$

discovering heavy hitters as well as their frequencies, it has been a challenge that mining frequent itemsets from all sanitized data. To the best of our konwledge, only SVSM [2] protocol that implements effectively FIM task in context of LDP. Particularly, it focuses on mining $top-k$ most frequent length-α ($\alpha > 1$) itemsets while utilizing SVIM [2] protocol that obtains the frequent items (or length-1 itemsets) as initial condition to construct candidate set. Details as follows.

$Set-Value\ Item\ Mining\ (SVIM)$: SVIM is a PSFO protocol with the same problem setting as LDPMiner [1], which aimming at discovering the k frequent items with highest frequencies. It has four steps and splits the privacy budget ϵ into three parts.

1. Prune the Domain ($\epsilon_1 - LDP$)**.** The goal is to identify a candidate set for items so that it can reduce the domian size into $2k$, which is greatly less than the domain d. In this step, each user uses FO to perturb a randomly sampled item from his raw data. Then, the analyst estimates the frequency of each value in the domain and selects the $2k$ frequent items with highest frequencies. The analyst broadcasts the domain set S pruned to all users.

2. Size Estimation ($\epsilon_2 - LDP$)**.** To further estimate the frequency of the set S, it is necessary to approximate the distribution of the number of frequent items that users hold. Thus, the perturbing data of each user is the size of the raw data intersected with the set S. After the perturbing of users finishes, the alalyst estimates the length distribution and calculates the $\gamma = 90\%$ length L by

$$\frac{\sum_{i=1}^{L} \tilde{\theta}(i)}{\sum_{i=1}^{2k} \tilde{\theta}(i)} > \gamma \tag{5}$$

3. Frequencies Estimation ($\epsilon_3 - LDP$)**.** Once the analyst gets S and L, it will use PSFO protocol to precisely estimate the frequencies of the items in small domian S. That is, each user in this step first pads his pruned data, which is the raw data intersected with S, to length L, then utilizes FO protocol for perturbing the item that randomly choose to report.

Finally, the analyst estimates the frequency of each item in S from all sanitized data. To ensure the estimation is unbiased, the estimated frequency needs to be multiplied by the length L.

4. Estimation Update. Due to the length is the 90th percentile length, which may lead to an underestimate. In order to improve the accuracy of the estimations, an update factor $u(L)$ is defined for correcting this under estimation. Practically, in this step, every frequency estimated is multiplied with the update factor,

$$u(L) := \frac{\sum_{i=1}^{2k} \tilde{\theta}(i)}{\sum_{i=1}^{2k} \tilde{\theta}(i) - \sum_{i=L+1}^{2k} \tilde{\theta}(i)(i-L)}$$

where $\tilde{\theta}(i)$ is the length distribution of length i estimated from step two.

Additionally, it is important to note that this step is the post-processing (Theorem II.2) in differential privacy and does not consume the privacy budget because informations are obtained from previous steps and there is no user involved.

When all the steps are done, the k items with highest frequencies are selected with high confidence.

$Set-Value\ itemSet\ Mining\ (SVSM)$: As mentioned above, SVSM protocol needs to know the $top-k$ frequent items, denoted by S'. Then, to constructe candidate set, the guessing frequency of each exponentially possible itemset that made up of items in S' is calculated by (1), and the domain IS is the constructed set that selects $top-2k$ itemsets with highest guessing frequencies.

While the domain of candidate itemsets is pruned, it may use following steps (two to four) for mining frequent itemsets. In particular, every user's sensitive data in SVSM is the power set of his raw data, which consist of a set of itemsets.

IV. FPMINE: THE FP-TREE-BASED APPROACH

In this section, we first introduce a baseline FP-tree-based approach to discover $top-k$ most frequent itemsets under LDP. Then, we propose the fpmine approach by optimizing the details to meet better performance.

A. The baseline FP-tree-based approach

In the local setting, the main challenge is that the analyst is unable to access users' raw data so that it is rather difficult to accurately construct a FP-tree. In [8], the FP-tree is constructed by scanning each user's transaction and updating nodes with frequent items that user holds in a depth-first manner. We observe that it enables to construct the FP-tree in a breadth-first manner if the count of nodes in each layer is known, which may be implemented through the FO protocols. Inspired by this, we propose a breadth-first approach called BLmine for costructing FP-tree layer by layer under LDP setting. Specifically, let L_m is the longest transaction length, and $S' = \left[x^1 : \tilde{f}(x^1), x^2 : \tilde{f}(x^2), ..., x^k : \tilde{f}(x^k)\right]$ denote the sorted sequence of k frequent items as well as their frequencies, where $\tilde{f}(x^i)$ denote the frequency of a frequent item x^i and for any $i > j$, it has $\tilde{f}(x^i) \geq \tilde{f}(x^j)$.

The BLmine approach has two phase:

Phase I Preprocessing. All participating users first delete infrequent items in their transactions and then rearrange the rest in order of items in S', that is $x_1 \succ x_2 \succ \cdots \succ x_k$, where $x_i \succ x_{i+1}$ means that x_i is more frequent than x_{i+1}. For example, the frequent items in Table I is order by $c \succ f \succ a \succ b \succ p$ while the minimum support threshold is set

TABLE III
PREPROCESSED TRANSACTIONAL DATA.

TID	List of items
T01	c, f, a, p
T02	c, f, a, b
T03	f, b
T04	c, b, p
T05	c, f, a, p

to 3, and then the preprocessed transactional data is shown in Table III.

Additionally, after the preprocessing step finishes, when $k \ll d$, there are many infrequent items discarded, which reduces significantly the size of candidate sets and achieves good performance because of the Apriori property [10]: *only the length* $-\ \alpha$ *itemset is frequent, its length* $-\ (\alpha+1)$ *superset is likely to be frequent.*

Phase II Constructing the FP-tree In contrast to FP-growth, we propose a breadth-first approach to construct the FP-tree layer by layer in the local setting. Algorithm 1 introduces the specific process in detail.

For example, lets consider the processed transactions in Table III, where the frequnet items and their supports is denoted by $S' = [c:4, f:4, a:3, b:3, p:3]$ (for simplicity, the frequency is replaced by the support here). Then, the root node v_0 is initialized firstly as the 0-th level of the tree and the constructing process is as follows.

In the bigining, add the children v_1, v_2, v_3, v_4, v_5 of v_0 to form the candidate node at the 1-st level of the tree, which initialized each count is zero. And then according to the candidate prefix set $C_1 = \{(c),(f),(a),(b),(p)\}$, their informations are collected for updating counts and pruning nodes, that is, the counts of v_1, v_2 are updated to 4 and 1 while v_3, v_4, v_5 are pruned because of the invalid counts. After the 1-st level are cunstructed, the set $C_2 = \{(c,f),(c,a),(c,b),(c,p),(f,a),(f,b),(f,p)\}$ at the 2-nd level is computed. It can update the tree nodes in the same way while the transaction of each user is the first two items in his pre-processed data (e.g. the input value of first user is (c,f)). BSmine continues to the next iteration until the tree reaches its maximum level or there are no node is valid. Finally, the FP-tree is constructed as show in Fig. 1.

When the analyst gets the FP-tree, the origin mining method is used to mine frequent itemsets. Therefore, we omit the mining procedure and remark on the differences from FP-growth.

- A node v in the FP-tree has two fields: $v.item$ and $v.count$, where denote the indicated item of node v and the number of its prefix $pre(v)$ in transactions respectively.
- We split the privacy budget ϵ into three parts ϵ_1 and ϵ_2. And the informations of the S' and L_m are obtained in a way that satisfies the ϵ_1-LDP. The formar is used to arrange users' frequent items and the latter marks the end of constructing the FP-tree.
- We construct the FP-tree in a breadth-first manner, which is unlike to the original algorithm, that is, the information is collected for updating the tree nodes layer by layer. Notablely, the input value of each user in l-th level is either an element of the candidate prefix set C_l or the dummy value †. Therefore, we only use the existing FO protocol (e.g. OLH) with the finite domain $C_l \cup \dagger$ to collect information at each level.
- We randomly divide users into L_m equal-sized groups and users in each group use the full privacy budget ϵ. It turns out that the overall has better accuracy and satisfies ϵ_2-LDP as well ([11], [17]).
- Let X denote a discovered itemset and its estimated frequency $\tilde{\theta}(X)$ is unbiased. Note that $\tilde{\theta}(X)$ is obtained from m conditional patter bases $B_1, B_2, ..., B_m$. Unlike the original algorithm to add directly, the final $\tilde{\theta}(X)$ is calculated in a way that minimize its variance as in Theorem IV.1.
——!!Notice whether it's different from a linear combination!!——

Theorem IV.1 *Given m noisy counts $\tilde{\theta}(b_1), \tilde{\theta}(b_2), ..., \tilde{\theta}(b_m)$ with variances of $r_1, r_2, ..., r_m$. The variance of $X = \sum_{i=1}^m \omega_i \tilde{\theta}(b_i)$ with constrain $\sum_{i=1}^m \omega_i = m$ is minimized by setting $\omega_i = m \cdot \frac{r_i^{-1}}{\sum_{j=1}^m r_j^{-1}}$.*

Proof. *Obviously, with $\sum_{i=1}^m \omega_i = m$, finding the minimized variance of $X = \sum_{i=1}^m \omega_i \tilde{\theta}(b_i)$ is equivalent to solve*

$$\min \sum_{i=1}^m \omega_i r_i, s.t. \sum_{i=1}^m \omega_i = m \tag{6}$$

Then, we have the Lagrangian function

$$\mathcal{L} = \sum_{i=1}^m \omega_i r_i + B(m - \sum_{i=1}^m \omega_i)$$

Let $\frac{\partial \mathcal{L}}{\partial \omega_i} = 2r_i\omega_i - B$ be zero, we have $\omega_i = \frac{B}{2r_i}$. And according to the condition $\sum_{i=1}^m \omega_i = m$, we get the constant $B = \frac{2m}{\sum_{i=1}^m r_i^{-1}}$.

Finally, we have the optimal solution is $\omega_i = m \cdot \frac{r_i^{-1}}{\sum_{j=1}^m r_j^{-1}}$.

Lemma IV.1 *Algorithm 1 satisfies ϵ-LDP.*

Proof. *proof Algorithm satisfies LDP.*

Lemma IV.2 *For any node v in the tree, let $\tilde{\theta}(v)$ denote its estiamted count obtained by the algorithm 1 and $\theta(v)$ denote the ground truth. With at least $1-\beta$ probability, we have:*

$$|\tilde{\theta}(v) - \theta(v)| = O\left(\frac{L_m \cdots}{\epsilon}\right)$$

B. Narrowing the cadidate prefix set

We observer that the biggest limitation to its application is that the candidate prefix set C that identified during each iteration is necessary to gather informations. On the one hand, the set C determines the domain of the domain of

Algorithm 1 BLmine(ϵ, S', L_m)

Randomly and evenly divide all users into L_m groups $g_1, g_2, ..., g_{L_m}$ with $n_g = \lfloor \frac{n}{L_m} \rfloor$ users each;
Initialize the tree with a root node v_r;
Initialize the count of v_r is n_g and mark it as valid;
Preprocess transactions of all participating users;
for $l = 1$ to L_m **do**
  while there exists a valid node v and $v.count > 0$ **do**
    Initialize a candidate prefix set $C_l = \emptyset$;
    Mark v as unvalid;
    for each item $x \in S'$ and $v.item \succ x.item$ **do**
      Add a child v_c of v with item is x and count is 0;
      Mark v_c as valid and compute its prefix $pre(v_c)$;
      Add the $pre(v_c)$ to candidate prefix set C_l;
    end for
  end while
  if C_l is $\emptyset$ **then**
    Break;
  end if
  for each users in group g_l **do**
    Perturbe his first l items with OLH protocol for collecting and updating an estimation count $\tilde{\theta}(v)$ of each node v, whose prefix is in C_l;
  end for
end for
return The root node v_r.

OLH protocol, so its size directly affects the accuracy of the estimations; On the other hand, the computational cost of identifing C is expensive when it is large. Specifically, the size of C is linearly related to the number of valid nodes reserved in the upper level of the tree. This is beacause only valid nodes may generate children, and these children make up the next level of candidate nodes.

In view of this, we first set the threshold T to retain as many valid estiamtions of OLH as possible and remove invalid results. It is noted that T is tradeoff and the larger T, the fewer invalid results, but more valid values may be lost, which reduces the accuracy. In this paper, we set $T =$, which is the maximum error boundary of OLH.
—-introduce T—-

It is obvious that there are many redundant prefixes in C at each iteration. For example, at the 3-rd level of the tree in Fig. 1, the set C_3 is made up of prefixes of $v_6, v_7, v_8, v_9, v_{10}, v_{11}, v_{12}$ and v_7, v_9, v_{10}, v_{12} are then marked as invalid. Therefore, if we could remove as many of these as possible in advance, then the range will be significantly narrowed.

Similar to SVSM [2] is limited candidate set to a fixed size. We propose the NC method to further narrow the set C_i at i-th iteration down to fixed size $\xi \cdot k$, where ξ is a predefined adiustable parameter. Spacifically, we guess the frequency of candidate itemsets in C_i, and then select the $\xi \cdot k$ itemsets with highest guessing frequency. Details in algorithm 2.

Algorithm 2 NC(S', C_i)

Initialize d_c is the size of C_i;
Initialize $C_i' = \emptyset$;
if $d_c \leq \xi k$ **then**

end if
return C_i'

C. *Imposing consistency and weight*

V. Ease of Use

A. *Maintaining the Integrity of the Specifications*

The IEEEtran class file is used to format your paper and style the text. All margins, column widths, line spaces, and text fonts are prescribed; please do not alter them. You may note peculiarities. For example, the head margin measures proportionately more than is customary. This measurement and others are deliberate, using specifications that anticipate your paper as one part of the entire proceedings, and not as an independent document. Please do not revise any of the current designations.

VI. Prepare Your Paper Before Styling

Before you begin to format your paper, first write and save the content as a separate text file. Complete all content and organizational editing before formatting. Please note sections VI-A–VI-E below for more information on proofreading, spelling and grammar.

Keep your text and graphic files separate until after the text has been formatted and styled. Do not number text heads—LaTeX will do that for you.

A. *Abbreviations and Acronyms*

Define abbreviations and acronyms the first time they are used in the text, even after they have been defined in the abstract. Abbreviations such as IEEE, SI, MKS, CGS, ac, dc, and rms do not have to be defined. Do not use abbreviations in the title or heads unless they are unavoidable.

B. *Units*

- Use either SI (MKS) or CGS as primary units. (SI units are encouraged.) English units may be used as secondary units (in parentheses). An exception would be the use of English units as identifiers in trade, such as “3.5-inch disk drive”.
- Avoid combining SI and CGS units, such as current in amperes and magnetic field in oersteds. This often leads to confusion because equations do not balance dimensionally. If you must use mixed units, clearly state the units for each quantity that you use in an equation.
- Do not mix complete spellings and abbreviations of units: “Wb/m^2” or “webers per square meter”, not “webers/m^2”. Spell out units when they appear in text: “. . . a few henries”, not “. . . a few H”.
- Use a zero before decimal points: “0.25”, not “.25”. Use “cm^3”, not “cc”.)

C. Equations

Number equations consecutively. To make your equations more compact, you may use the solidus (/), the exp function, or appropriate exponents. Italicize Roman symbols for quantities and variables, but not Greek symbols. Use a long dash rather than a hyphen for a minus sign. Punctuate equations with commas or periods when they are part of a sentence, as in:

$$a + b = \gamma \tag{7}$$

Be sure that the symbols in your equation have been defined before or immediately following the equation. Use "(7)", not "Eq. (7)" or "equation (7)", except at the beginning of a sentence: "Equation (7) is . . ."

D. LaTeX-Specific Advice

Please use "soft" (e.g., `\eqref{Eq}`) cross references instead of "hard" references (e.g., `(1)`). That will make it possible to combine sections, add equations, or change the order of figures or citations without having to go through the file line by line.

Please don't use the `{eqnarray}` equation environment. Use `{align}` or `{IEEEeqnarray}` instead. The `{eqnarray}` environment leaves unsightly spaces around relation symbols.

Please note that the `{subequations}` environment in LaTeX will increment the main equation counter even when there are no equation numbers displayed. If you forget that, you might write an article in which the equation numbers skip from (17) to (20), causing the copy editors to wonder if you've discovered a new method of counting.

BibTeX does not work by magic. It doesn't get the bibliographic data from thin air but from .bib files. If you use BibTeX to produce a bibliography you must send the .bib files.

LaTeX can't read your mind. If you assign the same label to a subsubsection and a table, you might find that Table I has been cross referenced as Table IV-B3.

LaTeX does not have precognitive abilities. If you put a `\label` command before the command that updates the counter it's supposed to be using, the label will pick up the last counter to be cross referenced instead. In particular, a `\label` command should not go before the caption of a figure or a table.

Do not use `\nonumber` inside the `{array}` environment. It will not stop equation numbers inside `{array}` (there won't be any anyway) and it might stop a wanted equation number in the surrounding equation.

E. Some Common Mistakes

- The word "data" is plural, not singular.
- The subscript for the permeability of vacuum μ_0, and other common scientific constants, is zero with subscript formatting, not a lowercase letter "o".
- In American English, commas, semicolons, periods, question and exclamation marks are located within quotation marks only when a complete thought or name is cited, such as a title or full quotation. When quotation marks are used, instead of a bold or italic typeface, to highlight a word or phrase, punctuation should appear outside of the quotation marks. A parenthetical phrase or statement at the end of a sentence is punctuated outside of the closing parenthesis (like this). (A parenthetical sentence is punctuated within the parentheses.)
- A graph within a graph is an "inset", not an "insert". The word alternatively is preferred to the word "alternately" (unless you really mean something that alternates).
- Do not use the word "essentially" to mean "approximately" or "effectively".
- In your paper title, if the words "that uses" can accurately replace the word "using", capitalize the "u"; if not, keep using lower-cased.
- Be aware of the different meanings of the homophones "affect" and "effect", "complement" and "compliment", "discreet" and "discrete", "principal" and "principle".
- Do not confuse "imply" and "infer".
- The prefix "non" is not a word; it should be joined to the word it modifies, usually without a hyphen.
- There is no period after the "et" in the Latin abbreviation "et al.".
- The abbreviation "i.e." means "that is", and the abbreviation "e.g." means "for example".

An excellent style manual for science writers is [24].

F. Authors and Affiliations

The class file is designed for, but not limited to, six authors. A minimum of one author is required for all conference articles. Author names should be listed starting from left to right and then moving down to the next line. This is the author sequence that will be used in future citations and by indexing services. Names should not be listed in columns nor group by affiliation. Please keep your affiliations as succinct as possible (for example, do not differentiate among departments of the same organization).

G. Identify the Headings

Headings, or heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads.

Component heads identify the different components of your paper and are not topically subordinate to each other. Examples include Acknowledgments and References and, for these, the correct style to use is "Heading 5". Use "figure caption" for your Figure captions, and "table head" for your table title. Run-in heads, such as "Abstract", will require you to apply a style (in this case, italic) in addition to the style provided by the drop down menu to differentiate the head from the text.

Text heads organize the topics on a relational, hierarchical basis. For example, the paper title is the primary text head because all subsequent material relates and elaborates on this one topic. If there are two or more sub-topics, the next level head (uppercase Roman numerals) should be used and, conversely, if there are not at least two sub-topics, then no subheads should be introduced.

H. Figures and Tables

a) Positioning Figures and Tables: Place figures and tables at the top and bottom of columns. Avoid placing them in the middle of columns. Large figures and tables may span across both columns. Figure captions should be below the figures; table heads should appear above the tables. Insert figures and tables after they are cited in the text. Use the abbreviation "Fig. 2", even at the beginning of a sentence.

TABLE IV
TABLE TYPE STYLES

Table Head	**Table Column Head**		
	Table column subhead	***Subhead***	***Subhead***
copy	More table copy[a]		

[a]Sample of a Table footnote.

Fig. 2. Example of a figure caption.

Figure Labels: Use 8 point Times New Roman for Figure labels. Use words rather than symbols or abbreviations when writing Figure axis labels to avoid confusing the reader. As an example, write the quantity "Magnetization", or "Magnetization, M", not just "M". If including units in the label, present them within parentheses. Do not label axes only with units. In the example, write "Magnetization (A/m)" or "Magnetization {A[m(1)]}", not just "A/m". Do not label axes with a ratio of quantities and units. For example, write "Temperature (K)", not "Temperature/K".

ACKNOWLEDGMENT

The preferred spelling of the word "acknowledgment" in America is without an "e" after the "g". Avoid the stilted expression "one of us (R. B. G.) thanks ...". Instead, try "R. B. G. thanks...". Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

Please number citations consecutively within brackets [18]. The sentence punctuation follows the bracket [19]. Refer simply to the reference number, as in [20]—do not use "Ref. [20]" or "reference [20]" except at the beginning of a sentence: "Reference [20] was the first ..."

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

Unless there are six authors or more give all authors' names; do not use "et al.". Papers that have not been published, even if they have been submitted for publication, should be cited as "unpublished" [21]. Papers that have been accepted for publication should be cited as "in press" [22]. Capitalize only the first word in a paper title, except for proper nouns and element symbols.

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [23].

REFERENCES

[1] Qin, Zhan, et al. "Heavy Hitter Estimation over Set-Valued Data with Local Differential Privacy." computer and communications security (2016): 192-203.
[2] Wang, Tianhao, Ninghui Li, and Somesh Jha. "Locally Differentially Private Frequent Itemset Mining." ieee symposium on security and privacy (2018): 127-143.
[3] Bhaskar, Raghav, et al. "Discovering frequent patterns in sensitive data." knowledge discovery and data mining (2010): 503-512.
[4] Li, Ninghui, et al. "PrivBasis: frequent itemset mining with differential privacy." very large data bases (2012): 1340-1351.
[5] Lee, Jaewoo, and Chris Clifton. "Top-k frequent itemsets via differentially private FP-trees." knowledge discovery and data mining (2014): 931-940.
[6] Zeng, Chen, Jeffrey F. Naughton, and Jinyi Cai. "On differentially private frequent itemset mining." very large data bases (2012): 25-36.
[7] C. Dwork. Differential privacy. In ICALP, pages 1–12, 2006.
[8] Han J, Pei J, Yin Y, et al. Mining frequent patterns without candidate generation[C]. international conference on management of data, 2000, 29(2): 1-12.
[9] Agrawal R, Imielinski T, Swami A N, et al. Mining association rules between sets of items in large databases[C]. international conference on management of data, 1993, 22(2): 207-216.
[10] Agrawal R, Srikant R. Fast Algorithms for Mining Association Rules in Large Databases[C]. very large data bases, 1994: 487-499.
[11] Wang T, Blocki J, Li N, et al. Locally Differentially Private Protocols for Frequency Estimation[C]. usenix security symposium, 2017: 729-745.
[12] F. D. McSherry. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In SIGMOD, pages 19–30, 2009.
[13] Zaki M J. Scalable algorithms for association mining[J]. IEEE Transactions on Knowledge and Data Engineering, 2000, 12(3): 372-390.
[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in Theory of Cryptography Conference. Springer, 2006, pp. 265–284.
[15] S. L. Warner. Randomized response: A survey technique for eliminating evasive answer bias. Journal of the American Statistical Association, 60(309):63–69, 1965.
[16] U. Erlingsson, V. Pihur, and A. Korolova. Rappor: Randomized aggregatable privacy-preserving ordinal response. In CCS, pages 1054–1067. ACM, 2014.
[17] Wang N, Xiao X, Yang Y, et al. PrivTrie: Effective Frequent Term Discovery under Local Differential Privacy[C]. international conference on data engineering, 2018: 821-832.
[18] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
[19] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.
[20] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[21] K. Elissa, "Title of paper if known," unpublished.
[22] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
[23] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[24] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.