Since January 2020 Elsevier has created a COVID-19 resource centre with free information in English and Mandarin on the novel coronavirus COVID-19. The COVID-19 resource centre is hosted on Elsevier Connect, the company's public news and information website.

Elsevier hereby grants permission to make all its COVID-19-related research that is available on the COVID-19 resource centre - including this research content - immediately available in PubMed Central and other publicly funded repositories, such as the WHO COVID database with rights for unrestricted research re-use and analyses in any form or by any means with acknowledgement of the original source. These permissions are granted for free by Elsevier for as long as the COVID-19 resource centre remains active.
Locally private frequency estimation of physical symptoms for infectious disease analysis in Internet of Medical Things

Xiaotong Wu
Department of Computer Engineering, Persian Gulf University, Bushehr, Iran

Mohammad Reza Khosravi
School of Information, Science and Engineering, Qufu Normal University, PR China

Lianyong Qi
Department of Computer Engineering, Persian Gulf University, Bushehr, Iran

Genlin Ji
School of Computer Science and Technology, Nanjing Normal University, PR China

Wanchun Dou
School of Computer and Software, Nanjing University, PR China

Xiaolong Xu
School of Computer Science and Technology, Nanjing University, PR China

A R T I C L E   I N F O

Keywords:
Health privacy
Frequency estimation
Local differential privacy
Infectious disease analysis

A B S T R A C T

Frequency estimation of physical symptoms for peoples is the most direct way to analyze and predict infectious diseases. In Internet of medical Things (IoMT), it is efficient and convenient for users to report their physical symptoms to hospitals or disease prevention departments by various mobile devices. Unfortunately, it usually brings leakage risk of these symptoms since data receivers may be untrusted. As a strong metric for health privacy, local differential privacy (LDP) requires that users should perturb their symptoms to prevent the risk. However, the widely-used data structure called sketch for frequency estimation does not satisfy the specified requirement. In this paper, we firstly define the problem of frequency estimation of physical symptoms under LDP. Then, we propose four different protocols, i.e., CMS-LDP, FCS-LDP, CS-LDP and FAS-LDP to solve the above problem. Next, we demonstrate that the designed protocols satisfy LDP and unbiased estimation. We also present two approaches to implement the key component (i.e., universal hash functions) of protocols. Finally, we conduct experiments to evaluate four protocols on two real-world datasets, representing two different distributions of physical symptoms. The results show that CMS-LDP and CS-LDP have relatively optimal utility for frequency estimation of physical symptoms in IoMT.

1. Introduction

With the explosive development of Internet of Medical Things (IoMT), there have been various medical applications and services in a large number of mobile devices (e.g., smart phones and wearable devices) [1–3]. It is efficient and convenient for hospitals or disease prevention departments (i.e., the third party) to estimate the frequency of physical symptoms for mobile users by their devices so as to monitor and predict infectious diseases. For example, if disease prevention departments want to know potential spread range and speed of coronavirus disease 2019 (COVID-19) [4], it is secure for them to remotely estimate the frequency of typical symptoms of peoples by smart phones, including fever, cough and shortness of breath. The novel estimation model reduces the labor costs and improves the detection efficiency. Although human beings benefit from the estimation model in IoMT, users inevitably face leakage risks of disease information, especially when the third party are untrusted [5,6]. Once disease information is leaked, it may cause a heavier psychological burden for the society and individuals.

In fact, health privacy has been one of the biggest concerns of not only individuals but also the whole society (e.g., European GDPR law) [7–12]. Therefore, it is necessary to protect individuals' health privacy when collecting and analyzing their symptoms. Differential privacy (DP) [13,14] is a strong privacy metric in a central setting. It assumes that there is a trusted third party to collect and perturb symptoms from users and then share them with others. However, DP ignores that it is possible for the third party to threaten users' health privacy, especially in IoMT. To this end, local differential privacy (LDP) [15] is proposed in the local setting. It requires that each mobile user should locally perturb his/her symptoms before sending to the third party and make them indistinguishable.

In recent years, there have been a series of works to design various protocols for frequency estimation of physical symptoms under LDP in both academia and industry [16–19]. In the existing privacy protocols, physical symptom is defined as categorical data. For categorical data, it consists of three important components, i.e., encoding, perturbation and...
aggregation [20–24]. The first two components are operated by the user while the last one is executed by the third party. Most of perturbation mechanisms to guarantee privacy are based on the Randomized Response (RR) technique, in which reporting a real answer for a Boolean query is derived from a certain probability [25]. Wang et al. [24] surveyed the previous protocols and classified encoding methods into four types, including direct [26], histogram [22], unary [23,27] and local hashing [20,21,24] encoding (i.e., DE, HE, UE and LHE). Protocols based on UE need higher communication cost, while those based on DE and HE have relatively lower accuracy. As a result, protocols based on LHE are a promising solution to balance computation complexity and data utility.

However, the previous protocols do not make full use of the widely applied data structures for frequency estimation of physical symptoms. Sketch is one of the most fundamental and most efficient data structures to estimate frequency for physical symptoms. In general, it uses a small vector to index the position of a symptom by a certain number of hash functions, which saves space but faces possible conflict. There are some basic and classic sketches, such as Count Sketch (CS) [28], Count-Min Sketch (CMS) [29], Fast-AGMS Sketch (FAS) [30] and Fast-Count Sketch (FCS) [31]. The focus of the paper is on how to design the sketch-based protocols for frequency estimation of physical symptoms satisfying local differential privacy. If so, it is convenient for others to directly apply the proposed protocols to estimate the frequency of physical symptoms.

There are two main challenges to solve the above problem. The first one is that the existing private protocols cannot be directly used to perturb query result by sketches. The reasons are that (i) perturbation and aggregation of symptoms are executed by the user and the third party, respectively. It implies that only the RR technique can be leveraged; and (ii) hash functions used in the sketches need to stay the same implying that the existing encoding methods (e.g., DE, HE and UE) are not applicable. The second one is that the designed private sketches should satisfy two important properties, including unbiased estimation and local differential privacy. In other words, it needs to balance privacy guarantee and utility of perturbed data.

To address the above challenges, we make full use of LHE and RR. In brief, we leverage the sketch vector to implement encoding of physical symptoms and then perturb it by RR to achieve privacy guarantee. At first, we propose two CMS- and FCS-based protocols, namely CMS-LDP and FCS-LDP to guarantee privacy. Then, we propose two CS- and FAS-based protocols, namely CS-LDP and FAS-LDP to ensure privacy. In particular, the formers utilize perturbation mechanism of optimal local hashing (OLH) in [24], while the latter directly use RR to perturb each entry of the sketch vector. Finally, we do other operations to make sure that the protocols satisfy unbiased estimation. It is noted that our protocols only increase computation cost rather than space overhead. This is consistent with the initial purpose, i.e., space save. To the best of our knowledge, there are few works to utilize the sketch under LHE to estimate the frequency of symptoms. The contributions of this paper can be summarized as follows:

- We formulate the problem of designing sketches under LDP by mathematical definitions and strong privacy metric to estimate the frequency of physical symptoms in IoMT.
- We propose four different protocols for frequency estimation of symptoms with privacy protection, including CMS-LDP, FCS-LDP, CS-LDP and FAS-LDP. We also demonstrate that all of the protocols satisfy two important properties, including unbiased estimation and LDP. We also analyze the variance of frequency estimation in four protocols.
- We present two approaches to implement universal hash functions. We also evaluate and compare the proposed protocols on two real-world datasets, representing two different distributions of physical symptoms.

The rest of this paper is organized as follows. Section 2 surveys related work about privacy protection in IoMT. Section 3 introduces the preliminaries and gives problem definitions. Section 4 proposes four different protocols for sketches under LDP to estimate the frequency of symptoms. Section 5 analyzes unbiased estimation, privacy guarantee and variance. Section 6 presents two methods to implement universal hash functions. Section 7 evaluates proposed protocols on real datasets. Finally, Section 8 concludes this paper.

2. Related work

2.1. Health privacy protection in IoMT

In IoMT, there are vast amounts of users’ healthy information generated by various mobile devices every day. However, the information is left from mobile users and may be faced with leakage risk from untrusted data collectors or data adversaries [6]. In order to guarantee privacy, there have been a number of protection techniques, including identification, anonymity and differential privacy [5,7,14]. In detail, identification technique utilizes authentication mechanism for each mobile user, who must login the system by his/her identity. On the other hand, data collectors use perturbation techniques (e.g., k-anonymity [32], l-diversity [33] and DP [13]) to perturb raw data from mobile users. These techniques provide a certain degree of privacy protection in some specified scenes.

However, these techniques are not fully suited to applications in IoMT due to their respective shortcomings. Identification and anonymity cannot offer enough privacy protection, while DP usually assumes that the third party is trusted. That is, the third party does not illegally leak users’ health information. Unfortunately, it does not hold in some real environments of IoMT. The reason is that users cannot know the receivers of their healthy information in the local setting so that they do not trust them. In order to overcome the disadvantages, local differential privacy [15] has been proposed to protect private information of each user against the threat from the untrusted third party. Both academic researchers and industrial engineers focus on how to achieve local differential privacy from theory and application.

2.2. Local differential privacy

Different from DP, LDP has higher privacy requirement in a local setting, which requires probability limit between any two values, instead of any two datasets in DP [15]. To this end, the existing mechanisms of DP cannot be fully applied to LDP, including the Laplace mechanism [34] and the exponential mechanism [35]. In order to implement LDP, there have been a number of works to design proper mechanisms [16–19]. Most of them leveraged the Randomized Response (RR) mechanism [25] or its refinements. RR reports a random answer derived from a certain amount of probability for a Boolean question. For categorical data, Kairouz et al. [36,37] proposed an extremal privatization mechanism (k-RR), which is universally optimal in the low and high privacy regimes. For ordinal data, the value is firstly encoded to a binary one and then perturbed by RR [17,38,39]. In general, the proposed algorithms need to satisfy both unbiased estimation and LDP. In addition, these perturbation privacy mechanisms are widely applied to various scenes, including heavy hitter identification, itemset mining, marginal release, data mining, medical analysis and infectious disease prediction [40].

In the industry, there have been a lot of applications to satisfy local differential privacy. RAPPOR is the first product from Google to collect users’ data and ensure privacy [27]. The user firstly encodes his/her data by a special data structure called Bloom filter and then perturbs the encoded output by the randomized response mechanism. The third party collects and decodes perturbed data from the users to get statistical information. In 2017, Apple [41] published a white paper to propose efficient and scalable algorithms satisfying local differential privacy. Microsoft [38] also presented new algorithms with local differential privacy to estimate mean and histogram for telemetry data.
Each symptom is labeled as an integer (e.g., fever is labeled by 1), symptoms of COVID-19, such as fever, cough and shortness of breath.

### 3. Preliminaries and problem definition

As aforementioned, the focus of this paper is to solve the problem of frequency estimation of physical symptoms in IoMT. Therefore, the main objective of the existing mechanisms is to maximize the utility and reduce communication cost and computation complexity.

Bassily et al. [21] proposed efficient mechanisms by utilizing Random Matrix Projection. Running time of the third party and the user is $O(n^{3/2})$ and $O(n^{1/2})$, respectively. Meanwhile, the estimation error is $O(\sqrt{\log(n)}/(c^2n))$ with high probability where $n$ is the number of users and $m$ is the size of domain. They also proposed improved algorithms, in which server time is $O(n)$ and user time is $O(1)$ [20].

Bun et al. [22] presented an algorithm called PrivateExpanderSketch, which achieved optimal worst-case error as a function of all possible parameters. Wang et al. [24] introduced a framework to analyze all the previous mechanisms for frequency estimation under LDP. They also proposed an Optimal Local Hashing (OLH) to estimate the frequency with better utility. Different from the previous environment with only an attribute, Qin et al. [23] estimated frequency over set-valued data and for any pair of input $d \in D$, $\hat{f}(d)$ satisfies the following requirement:

$$\Pr \left[ |f(d) - \hat{f}(d)| \leq \epsilon d^2 \right] \geq 1 - \delta,$$

(2)

where $\xi \in [0, 1]$ represents the expected error and $\delta \in [0, 1]$ is the confidence probability.

Unfortunately, the third party may be untrusted because that he/she possibly utilizes health information of users to get illegal profit. Therefore, the users have to perturb their information so as to avoid privacy leakage. Differential privacy [13,14] is the most common and strong privacy protection metric, which is usually suited for the central setting. Different from DP, local differential privacy is proposed in the local setting, in which it assumes that the third party is untrusted. This means that before the users send their symptoms, perturbation of symptoms should satisfy the following privacy measurement:

### Definition 2 ((\xi, \delta)-Approximate Frequency Estimation of Symptoms [39])

A query $f : D \rightarrow \mathbb{Z}$ is to solve the problem of $(\xi, \delta)$-approximate frequency estimation of symptoms if and only if for any symptom $d \in D$, $\hat{f}(d)$ satisfies the following requirement:

$$\Pr \left[ |f(d) - \hat{f}(d)| \leq \epsilon d^2 \right] \geq 1 - \delta,$$

(2)

where $\xi \in [0, 1]$ represents the expected error and $\delta \in [0, 1]$ is the confidence probability.

A sanitized algorithm $M$ satisfies $\epsilon$-local differential privacy if and only if for any pair of input values $d_1, d_2 \in D$ and for any output $y \in \text{Range}(M)$, we have

$$\Pr[M(d_1) = y] \leq \exp(\epsilon) \cdot \Pr[M(d_2) = y].$$

(3)

where $\text{Range}(M)$ denotes the set of all possible outputs of the algorithm $M$ and $\epsilon$ is privacy budget. $\epsilon$ represents the strength of privacy protection. The lower value of $\epsilon$ corresponds to the stronger privacy protection. With the same as DP, LDP has also the composition properties as follows.

**Lemma 1 (Sequential Composition [43])**. Assume that each privacy algorithm $M_i$ satisfies $\epsilon_i$-local differential privacy. A group of $M_i(\cdot)$ applied to the same dataset satisfy $\left( \sum_i \epsilon_i \right)$-local differential privacy.

**Lemma 2 (Parallel Composition [43])**. Assume that each privacy algorithm $M_i$ satisfies $\epsilon_i$-local differential privacy. A group of $M_i(\cdot)$ applied to a set of disjoint datasets satisfy $\max_i \epsilon_i$-local differential privacy.

In the algorithms designed by the paper, we will make full use of the composition properties of local differential privacy.

**Definition 3 (Local Differential Privacy [15])**. A sanitized algorithm $M$ satisfies $\epsilon$-local differential privacy if and only if for any pair of input values $d_1, d_2 \in D$ and for any output $y \in \text{Range}(M)$, we have

$$\Pr[M(d_1) = y] \leq \exp(\epsilon) \cdot \Pr[M(d_2) = y].$$

(3)

where $\text{Range}(M)$ denotes the set of all possible outputs of the algorithm $M$ and $\epsilon$ is privacy budget. $\epsilon$ represents the strength of privacy protection. The lower value of $\epsilon$ corresponds to the stronger privacy protection. With the same as DP, LDP has also the composition properties as follows.

**Lemma 1 (Sequential Composition [43])**. Assume that each privacy algorithm $M_i$ satisfies $\epsilon_i$-local differential privacy. A group of $M_i(\cdot)$ applied to the same dataset satisfy $\left( \sum_i \epsilon_i \right)$-local differential privacy.

**Lemma 2 (Parallel Composition [43])**. Assume that each privacy algorithm $M_i$ satisfies $\epsilon_i$-local differential privacy. A group of $M_i(\cdot)$ applied to a set of disjoint datasets satisfy $\max_i \epsilon_i$-local differential privacy.
3.2. Private framework

In order to solve the frequency estimation problem under LDP for symptoms, the proposed mechanism should satisfy the privacy measurement as follows:

Definition 4 (Frequency Estimation of Symptoms under LDP). A sanitized protocol \( M : D^n \rightarrow Z \) for frequency estimation of symptoms satisfies \( \epsilon \)-local differential privacy if and only if \( M = M_1(M_2(\cdots M_n(\cdots))) \), in which each \( M_i : D \rightarrow Y \) is a local random processor that satisfies \( \epsilon \)-local differential privacy, and \( A : Y^n \rightarrow Z \) is some post-processing function.

Based on the above definition, the key functions are the local random processor \( M_i(\cdot) \) and the post-processing function \( A(\cdot) \). Each user executes the local random processor, while the third party executes the post-processing function. Furthermore, Wang et al. [24] surveyed the previous work and concluded that a protocol for frequency estimation under LDP generally consists of the following steps:

- **Encoding.** This operation Encode(\( \cdot \)) is executed by the user. In detail, input \( d \) of the user is generally encoded as a histogram, a vector [27], a single bit [21], or an integer [24] by encoding and decoding algorithms or hash functions.

- **Perturbing.** The mechanism Perturb(\( \cdot \)) is also executed by the user. Meanwhile, the perturbed mechanism should satisfy \( \epsilon \)-local differential privacy. The input of Perturb(\( \cdot \)) is the output of Encode(\( \cdot \)). Therefore, \( M_i(\cdot) \) in Definition 4 is Perturb(Encode(\( \cdot \))).

- **Aggregating.** The operation Aggregate(\( \cdot \)) is executed by the third party. At first, the third party receives data from users, which is perturbed with the combination of Encode(\( \cdot \)) and Perturb(\( \cdot \)). Then, he/she aggregates the data and extracts approximate frequency estimation.

Fig. 2 shows the concrete operations and interactions between mobile users and the third party. It is obvious to see that users need to encode and perturb their symptoms locally, while the third party aggregates perturbed data. As a result, these operations increase extra overhead for users, including computation, communication and storage. To this end, it is necessary to make full use of the existing data structure named sketch to decrease the overhead.

3.3. Problem formulation

In order to solve the approximate frequency estimation of categorical data (e.g., physical symptoms), Charikar et al. [28] firstly proposed an algorithm by using a special data structure, i.e., Count Sketch, which achieved good space bounds. After that, the refined versions were proposed, such as Fast-AGMS Sketch [30], Count-Min Sketch [29] and Fast-Count Sketch [31]. However, to the best of our knowledge, there are few works to design the new data structure Sketch under LDP to solve the approximate frequency estimation of physical symptoms. In the following sections, we propose improved protocols based on Count Sketch, Fast-AGMS Sketch, Count-Min Sketch and Fast-Count Sketch, which satisfy unbiased estimation, i.e., \( \forall d \in D, E[\hat{f}(d)] = f(d) \) and local differential privacy. In addition, we also compare the proposed protocols by experiments.

4. Protocols for frequency estimation of symptoms under LDP

In this section, we propose four novel protocols to solve the approximate frequency estimation of physical symptoms under LDP. These protocols are based on CS, FAS, CMS and FCS. We will analyze unbiased estimation and privacy protection of proposed protocols in the next section.

Under no privacy preservation, the third party receives physical symptoms and uses a small vector to count the frequency of each element in \( D \), while users do nothing. However, in order to protect privacy, the user should send perturbed symptoms rather than raw values. Therefore, the user needs to generate a small vector locally and then send it to the third party. Finally, the third party aggregates these vectors and derives the full sketch.

4.1. Protocol CMS-LDP

CMS is a simple and effective data structure, which was proposed by Cormode et al. [29]. For a frequency estimation query \( \hat{f}(d) \), its values by CMS is \( [\hat{f}(d), f(d)+\xi\lfloor f/d \rfloor] \) with a probability \( 1-\delta \), in which \( f/d \) is the real frequency except \( d \). Here, we propose corresponding protocols based on optimal local hashing in [24] to implement CMS under local differential privacy.

Algorithm 1 shows a protocol CMS-LDP for privacy estimation of symptoms under LDP. The protocol consists of four key components. The first one is an initial process, in which \( b \) and \( t \) are computed by the input parameters \( \xi \) and \( \delta \) (line 1 \~ 3). \( b \) represents the domain of universal hash functions, while \( t \) is the number of hash functions. Then, each user calls \( M_{\text{client-CMS-LDP}} \) to report the perturbed record to the server-side (line 4 \~ 6). Next, Line 7 implements the key component, namely Sketch-CMS-LDP. The last one implements the query of server-side (line 8 \~ 10). By the above four parts, the algorithm solves the approximate frequency estimation problem of physical symptoms.

Each user executes Algorithm 2 to implement the function of Perturb(Encode(\( \cdot \))) in Section 3.2. Algorithm 2 utilizes an existing technique, namely \( b \)-Randomized Response (\( b \)-RR), to ensure \( \epsilon \)-local differential privacy, which was designed by Kairouz et al. [37]. In detail,
Algorithm 3 Sketch-CMS-LDP \( M_{\text{Sketch-CMS-LDP}} \)

**Input:** Dataset \( D = \{V_1, V_2, \ldots, V_n\} \), privacy parameter \( \epsilon \), dimensions \( t \) and \( b \).

**Output:** sketch
1: \( S \leftarrow t \times b \)-integer array, initially 0
2: for \( i \in [t] \) do
3: for \( j \in [b] \) do
4: \( V[i][j] = V[i][j] + V[j] \)
5: end for
6: end for
7: return \( S \)

Algorithm 4 Server-Side \( M_{\text{server-CMS-LDP}} \)

**Input:** Symptom \( d \in D \), sketch \( S \), privacy parameter \( \epsilon \), hash functions \( H = \{h_1, h_2, \ldots, h_\ell\} \), the number of hash functions \( \ell \).

**Output:** frequency estimation \( f(d) \)
1: \( f(d) \leftarrow \min \{V[i][h_i(d)] : i \in [\ell]\} \)
2: return \( f(d) \)

Algorithm 2 firstly implements \( \text{Encode}(-) \) by universal hash function \( h_i \) to map the input \( d \) into a \( b \)-length bit array in line 4. Then, a \( b \)-RR technique is used to perturb the output of \( \text{Encode}(-) \) in line 5. Finally, the user returns the result to the server-side.

After the third party receives perturbed symptoms from users, he/she computes a data structure, namely Sketch-CMS-LDP by Algorithm 3. Compared to CMS, our proposed mechanism \( M_{\text{Sketch-CMS-LDP}} \) has extra operations, i.e., Line 7 ∼ 9. Sketch-CMS-LDP firstly generates an initial sketch, which is a multidimensional array in line 1. Then, all arrays from \( n \) users are added to the sketch in line 2 ∼ 6, which is the same with CMS. Finally, Line 7 ∼ 9 is to ensure unbiased estimation of CMS-LDP, which will be proved in Section 5.1.

Algorithm 4 gets the approximate estimation for each frequency query \( f(d) \), which is executed by the third party. The combination of Algorithms 3 and 4 implement the function of \( \text{Aggregate}(-) \) in Section 3.2. For each query \( f(d) \), the algorithm chooses the minimum of \( t \) values, each of which is from an array \( V[i][j] \). The position of each value in the array is decided by hash function \( h_i \) and \( d \). By Algorithm 1 ∼ 4, we solve the problem of frequency estimation under local differential privacy.

For each user in the client-side, the time complexity is \( O(\log_2(1/\delta)) \) and the space complexity in \( \text{Perturb(Encode(-))} \) is \( O(\frac{1}{\epsilon} \log_2(1/\delta)) \). For \( M_{\text{server-CMS-LDP}} \) in the server-side, the time complexity is \( O(\frac{1}{\epsilon} \log_2(1/\delta)) \) and the space complexity is \( O(\frac{1}{\epsilon} \log_2(1/\delta)) \). Besides, the communication overhead between each user and the server is \( O(\frac{1}{\epsilon} \log_2(1/\delta)) \). Thus, for the third party, our proposed protocol does not increase the space overhead. Section 5.1 will analyze the proposed algorithms in terms of the error bound, unbiased estimation and local differential privacy.

### 4.2. Protocol FCS-LDP

Thorup et al. [31] proposed a refined version of Count-Min Sketch, namely Fast-Count Sketch (FCS). Compared with CMS, FCS uses a family of 4-universal hash functions, instead of 2-universal hash function.

The protocol to implement Fast-Count Sketch under local differential privacy, namely Algorithm FCS-LDP, is similar with Algorithm 1. FCS-LDP also consists of three important components, i.e., \( M_{\text{client-FCS-LDP}} \), \( M_{\text{Sketch-FCS-LDP}} \) and \( M_{\text{server-FCS-LDP}} \). Different from CMS-LDP, FCS-LDP replaces 2-universal hash functions with 4-universal hash functions in line 3 in Algorithm 1. Besides, \( M_{\text{Sketch-FCS-LDP}} \), \( M_{\text{client-FCS-LDP}} \) and \( M_{\text{server-FCS-LDP}} \) have the same operations with Algorithm \( M_{\text{Sketch-CMS-LDP}} \), \( M_{\text{client-CMS-LDP}} \) and \( M_{\text{server-CMS-LDP}} \), respectively. Here, we do not repeat the same algorithms. Meanwhile, these algorithms have the same time and space complexity with Algorithm 1 ∼ 4 for each user and the third party.

### 4.3. Protocol CS-LDP

Among all of various formats of the special data structure called sketch, Count Sketch was firstly proposed in 2002 [28]. The data structure is generally used to point query, range query, inner product query and so on. Compared with Count-Min Sketch, CS needs higher space cost, i.e., \( O(\log_2(d)/\epsilon^2) \). Meanwhile, CS also satisfies unbiased estimation and the value for a frequency estimation query \( f(d) \) about \( d \) is \( [f(d) - 2\|f_d\|_2, f(d) + 2\|f_d\|_2] \) with a confidence probability \( 1 - \delta \).

In the following, we will introduce algorithms to implement CS with LDP.

In Algorithm 5, we describe the overview of protocol CS-LDP to implement CS under LDP. It consists of four key parts, i.e., initialization, perturbation in client-side, construction of sketch and frequency estimation in server-side. In the initial process (line 1 ∼ 4), it requires \( t \) hash function pairs \( (h_i, g_i) \). Hash function \( h_i \) maps some input into \( \lceil c/\epsilon^2 \rceil \)-length bits in which \( c \) is a constant, while function \( g_i \) maps some input value into \( [-1, +1] \). Both two classes of hash functions are independent with each other. Compared with \( [1/\epsilon] \)-length bits of hash function \( h_i \) in Algorithm 1, CS-LDP needs more space cost. Then, each user executes \( M_{\text{client-CS-LDP}} \) to perturb his/her data (line 5 ∼ 7), which implements the function of \( \text{Perturb(Encode(-))} \). The third party constructs a special data structure, namely Sketch-CS-LDP, to collect data from users by \( M_{\text{Sketch-CS-LDP}} \) (line 8). Finally, for each frequency estimation \( f(d) \), the third party calls \( M_{\text{server-CS-LDP}} \) to compute the result (line 9 ∼ 11).

Before each user sends his/her symptom to the third party, Algorithm 6 is executed. \( M_{\text{client-CS-LDP}} \) firstly initializes a multidimensional integer array \( V \), i.e., \( [1/\epsilon] \times d \) (line 1). The array only consists of two elements, i.e., \( -1 \) and \(+1\). Then, because of the property of sequential composition of local differential privacy, it computes smaller privacy parameter \( \epsilon' \) for each hash function (line 2). Next, the specified position \( h_i(d) \) of each row in \( V \) is set as \( g_i(d) \) (line 3 ∼ 5), which implements the function of \( \text{Encode}(-) \). For each entry in array \( V \), it uses Randomized Response (RR) [25] to ensure \( \epsilon' \)-local differential privacy (line 6 ∼ 10), which implements the function of \( \text{Perturb}(-) \). At last, the result is returned to the third party.

After the third party receives perturbed symptoms from users, he/she uses Algorithm 7 to get the data structure Sketch-CS-LDP. Firstly, \( M_{\text{Sketch-CS-LDP}} \) initializes a \( t \times b \)-integer array, i.e., \( [0/\epsilon] \times b \) (line 1). Then, line 2 is to compute a parameter \( c_{\epsilon\epsilon} \), which is used to ensure unbiased estimation due to randomized response. Since data from perturbed mechanism \( M_{\text{client-CS-LDP}} \) is biased, it needs to be calibrated. Therefore, Algorithm 7 adjusts the data from each user (line 3 ∼ 5) to get the estimated frequency of each entry in each array \( V \). Finally, all adjusted arrays are added to sketch (line 6 ∼ 9). In Section 5.2, we will prove unbiased estimation and local differential privacy of our proposed protocol CS-LDP.

Algorithm 8 is used to answer the frequency estimation \( f(d) \) for the third party. The combination of Algorithms 7 and 8 implements the function of \( \text{Aggregate}(-) \). We use function \( \text{median}(-) \) to choose a proper value from the candidates as the final estimation for \( d \), rather than function \( \text{mean}(-) \). Charikar et al. [28] has demonstrated that function \( \text{mean}(-) \) is sensitive to outliers, while \( \text{median}(-) \) is robust. By Algorithm 5 ∼ 8, we solve the problem of frequency estimation with local differential privacy.

Compared with CMS-LDP, CS-LDP needs the higher time and space cost, and has the sharper bound for the frequency estimation.
Algorithm 5 CS-based Estimation under LDP CS-LDP

Input: A set of symptoms $I = \{d_1, d_2, \ldots, d_n\} \in \mathcal{D}_t$, error parameter $\xi$, probability confidence $\delta$, privacy parameter $\epsilon$

Output: a set of approximate frequency estimation $\{\hat{f}(d)\}$

1: $b \leftarrow \lceil c/\xi^2 \rceil$ for some constant $c$
2: $t \leftarrow \lceil \log_2(1/\delta) \rceil$
3: choose a set $H = \{h_1, h_2, \ldots, h_t\}$ independently from a 2-universal family of hash functions mapping $D$ to $[b]$.
4: choose a set $\mathcal{G} = \{g_1, g_2, \ldots, g_t\}$ independently from a 2-universal family of hash functions mapping $D$ to $\{−1, +1\}$.
5: for $i \in [n]$ do
6: $Y_i \leftarrow \text{M}_\text{client-CS-LDP}(d_i, H, G, e, t, b)$
7: end for
8: Construct the sketch
   $Y \leftarrow \text{M}_\text{sketch-CS-LDP}((Y_1, Y_2, \ldots, Y_n), e, t, b)$
9: for $d \in \hat{D}$ do
10: $\hat{f}(d) \leftarrow \text{M}_\text{server-CS-LDP}(d, Y, e, H, G, t)$
11: end for
12: return $\{\hat{f}(d) : d \in \hat{D}\}$

Algorithm 6 Client-Side $\text{M}_\text{client-CS-LDP}$

Input: symptom $d_i$, hash functions $H = \{h_1, h_2, \ldots, h_t\}$ and $\mathcal{G} = \{g_1, g_2, \ldots, g_t\}$, privacy parameter $\epsilon$, the number of hash functions $t$, the bit number $b$

Output: matrix $Y$

1: $Y \leftarrow i \times b$-integer array, initially −1
2: $e' \leftarrow e/2t$
3: for $i \in [t]$ do
4: $Y[i][h_i(d)] = g_i(d)$
5: end for
6: for $i \in [t]$ do
7: for $j \in [b]$ do
8: sample some value $y$ as follows:
   $$y = \begin{cases} +1 & \text{w.p. } e' \\ -1 & \text{w.p. } e'e' + 1 \end{cases}$$
9: $Y[i][j] \leftarrow Y[i][j] \cdot y$
10: end for
11: end for
12: return $Y$

Algorithm 7 Sketch-CS-LDP $\text{M}_\text{sketch-CS-LDP}$

Input: Dataset $D = \{Y_1, Y_2, \ldots, Y_n\}$, privacy parameter $e$, dimensions $t$ and $b$

Output: sketch $Y$

1: $Y \leftarrow i \times b$-integer array, initially 0
2: $e' \leftarrow e/2\cdot2b$
3: for $i \in [t]$ do
4: $Y_i \leftarrow c_i \cdot Y_i$
5: end for
6: for $i \in [t]$ do
7: for $j \in [b]$ do
8: $Y[i][j] = Y[i][j] + Y_i[j]$
9: end for
10: end for
11: return $Y$

For each user in $\text{M}_\text{client-CS-LDP}$, the time cost is $O(\frac{t}{\epsilon^2} \log_2(1/\delta))$ due to sampling each entry of the matrix by Randomized Response, while the space cost is $\frac{t}{\epsilon^2} \log_2(1/\delta)$. For the third party, the cost of time and space is $O(\frac{t}{\epsilon^2} \log_2(1/\delta))$ and $O(\frac{t}{\epsilon^2} \log_2(1/\delta))$ in $\text{M}_\text{sketch-CS-LDP}$, respectively. The cost of time and space is $O(\log_2(1/\delta))$ and $O(\log_2(1/\delta))$ in $\text{M}_\text{server-CS-LDP}$, respectively. Therefore, the total time cost for the third party is $O(\frac{t}{\epsilon^2} \log_2(1/\delta))$, while the total space cost is $O(\frac{t}{\epsilon^2} \log_2(1/\delta))$.

4.4. Protocol FAS-LDP

Fast-AGMS Sketch (FAS) is a refined version of Count Sketch [30], which guarantees logarithmic-time sketch update and tracking costs. The only difference between Count Sketch and Fast-AGMS Sketch is that the latter replaces 2-universal hash functions $\mathcal{G}$ of the former with 4-universal hash functions.

Protocol FAS-LDP to achieve Fast-AGMS Sketch under local differential privacy is similar with CS-LDP. The only change is that a set of hash functions, namely $\mathcal{G}$, are chosen independently from a 4-universal family of hashing functions mapping $D$ to $\{−1, +1\}$. Besides, other mechanisms, i.e., $\text{M}_\text{client-FAS-LDP}$, $\text{M}_\text{sketch-FAS-LDP}$ and $\text{M}_\text{server-FAS-LDP}$ in client-side and server-side are the same with $\text{M}_\text{client-CS-LDP}$, $\text{M}_\text{sketch-CS-LDP}$ and $\text{M}_\text{server-CS-LDP}$, respectively. As a result, these algorithms have the same time and space cost with Algorithms 5–8 for each user and the third party.

5. Theory analysis of protocols

The protocols that implement the approximate frequency estimation of physical symptoms under LDP must satisfy two fundamental requirements, i.e., unbiased estimation and local differential privacy. In the section, we will demonstrate that our proposed protocols satisfy two requirements. Besides, we discuss the variance of protocols.

5.1. CMS-LDP and FCS-LDP

For the proposed protocols, the most important property is to satisfy local differential privacy. We prove that protocol CMS-LDP satisfies the requirement as follows:

**Theorem 3.** Protocol CMS-LDP satisfies $\epsilon$-local differential privacy.

**Proof.** Each user utilizes mechanism $\text{M}_\text{client-CMS-LDP}$ in CMS-LDP to encode and perturb his/her symptom $d$. According to Eq. 4, for two different value $d$ and $d'$, it satisfies the follow equation:

$$\Pr[\text{Perturb}(\text{Enc}(d)) = y] \leq e' + b - 1 + e' + b - 1$$

Therefore, for each hash function, perturbation satisfies $\epsilon'$-local differential privacy. Because that there are $t$ hash functions for the same symptom $d$, the combination of $t$ hash functions satisfy $\epsilon$-local differential privacy for each user according to Lemma 1. On the other hand, the third party receives perturbed symptoms from all users. Since records between different users are disjoint, it satisfies $\epsilon$-differential privacy for all perturbed symptoms according to Lemma 2. ☐
Theorem 4. For each 2-universal hash function $h_i$ in protocol CMS-LDP, it satisfies unbiased estimation for frequency estimation of symptoms after Encode(). Meanwhile, its variance is \[ \frac{(h(r)-hp)}{bp-1} \cdot f + \frac{(n(h(r)-1)}{bp-1} \cdot q, \] in which \( p = e^{\gamma} \) and \( q = e^{\gamma}. \)

Proof. For each hash function $h_i$, the third party receives $n$ reported symptoms from all users. Let $y$ any possible output and $S$ the set of symptoms that are hashed to $y$ by Encode(). Then, the probability that any symptom in $S$ is perturbed to $y$ is $p$. That is, for any $s_i \in S$, we have \[ p = \Pr[\text{Perturb}(\text{Encode}(s_i)) = y] = \frac{e^{\gamma}}{e^{\gamma} + b - 1}. \]

and \[ q = \Pr[\text{Perturb}(\text{Encode}(s_i)) \neq y] = \frac{1}{e^{\gamma} + b - 1}. \]

For another input value $d_j \notin S$, it is still possible mapped to $y$. Let $q'$ the probability that $d_j$ is mapped to $y$. In line 10 of mechanism $\mathcal{M}_{\text{Sketch-LMS-LDP}}$, the third party makes the extra operations for the raw records. Thus, the probability $q'$ is computed by as follows:

\[
q' = \frac{1}{b} \cdot b + \frac{b - 1}{b} \cdot q
= \frac{1}{b} \cdot \frac{e^{\gamma} + b - 1}{b} \cdot \frac{b - 1}{b} \cdot \frac{1}{e^{\gamma} + b - 1}
= \frac{1}{b}
\]

In mechanism $\mathcal{M}_{\text{Sketch-LMS-LDP}}$, we get the final sketch by line 11. The expectation that the number of elements in $D$ which are mapped to $y$ is

\[
E[f] = E \left[ \frac{f \cdot n \cdot q'}{p - q'} \right]
= E[f] - n \cdot q'
= p \cdot f + (n - f) \cdot q' - n \cdot q'
= f
\]

where $f$ is the number of elements in $S$. Therefore, for each 2-universal hash functions, protocol CMS-LDP satisfies unbiased estimation. The variance is

\[
\text{Var}[f] = \text{Var} \left[ \frac{f \cdot n \cdot q'}{p - q'} \right]
= \frac{(p - q')^2}{p - q'}
= \frac{f \cdot p(1 - p) + (n - f) \cdot q'(1 - q')}{(p - q')^2}
= \frac{f^2 p(1 - p) + (n - f)(b - 1)}{(b - 1)^2}
= \frac{(b - 1)(b - 1)}{b - 1} \cdot f + \frac{n(b - 1)}{(b - 1)^2}
\]

By the above proof, CMS-LDP satisfies unbiased estimation and LDP. Since protocol FCS-LDP is similar with CMS-LDP except for hash functions, FCS-LDP also satisfies unbiased estimation and local differential privacy. Besides, it has the same variance. The corresponding proof is omitted.

5.2. CS-LDP and FAS-LDP

Here, we firstly prove that protocol CS-LDP satisfies local differential privacy as follows:

Theorem 5. Protocol CS-LDP satisfies $\epsilon$-local differential privacy.

Proof. Assume that there are two users $u_i$ and $u_j$. Both of them have different input values $d_i$ and $d_j$ and the same output $O \in \{-1, +1\}^b$. For each pair $(h_i, g_i)$ of hash functions, the third party receives two perturbed arrays $\nu$ and $\omega$ from these two users. Because that all of hash functions $h_i$ and $g_i$ are independent with each other, we have

\[
\Pr[\mathcal{M}_{\text{Client-CS-LDP}}(d_i) = O] = \Pr[\mathcal{M}_{\text{Client-CS-LDP}}(d_j) = O]
= \prod_{k=1}^{b} \Pr[\nu_{x_k} = \Theta_{x_k}]
= \prod_{k=1}^{b} \Pr[\nu_{x_k} = \Theta_{x_k}] \quad \forall
\]

\[
\Pr[\mathcal{M}_{\text{Client-CS-LDP}}(d_i) = O_{h_i(d_i)}] \Pr[\nu_{h_i(d_i)} = \Theta_{h_i(d_i)}] = \Pr[\mathcal{M}_{\text{Client-CS-LDP}}(d_j) = O_{h_j(d_j)}] \Pr[\nu_{h_j(d_j)} = \Theta_{h_j(d_j)}] \Delta
\]

The reason for $\nu \Rightarrow \Delta$ is presented as follows. In detail, $\nu = \nu_1, \nu_2, \ldots, \nu_b$ and $\omega = \omega_1, \omega_2, \ldots, \omega_b$. Each entry $\nu_i$ in $\nu$ is derived from $[-1, +1, y]$, in which $y \in \{-1, +1\}$ is sampled from Eq. 6 and only an entry is $+1$ with position decided by hash value $h_i(d_i)$. $\omega$ is similar with $\nu$. In Eq. $\nu$, only two entries in $\nu_{h_i(d_i)}$ are different from two entries in $\omega_{h_i(d_i)}$, while the others have the same probability. Therefore, $\nu \Rightarrow \Delta$ is derived to $\Delta$. For Eq. $\Delta$, there are two different cases, i.e., $h_i(d_i) = h_j(d_j)$ and $h_i(d_i) \neq h_j(d_j)$. For the first case in which $h_i(d_i) = h_j(d_j)$, we have

\[
\Delta = \frac{\Pr[\nu_{h_i(d_i)} = \Theta_{h_i(d_i)}]}{\Pr[\omega_{h_i(d_i)} = \Theta_{h_i(d_i)}]} = \frac{\Pr[\nu_{h_i(d_i)} = \Theta_{h_i(d_i)}]}{\Pr[\omega_{h_i(d_i)} = \Theta_{h_i(d_i)}]} = 1
\]

For the second case in which $h_i(d_i) \neq h_j(d_j)$, we have

\[
\Delta = \frac{\Pr[\nu_{h_i(d_i)} = \Theta_{h_i(d_i)}]}{\Pr[\omega_{h_i(d_i)} = \Theta_{h_i(d_i)}]} = \frac{\Pr[\nu_{h_i(d_i)} = \Theta_{h_i(d_i)}]}{\Pr[\omega_{h_i(d_i)} = \Theta_{h_i(d_i)}]} = 1
\]

Assume that $g_i(d_i) = +1$ and $g_i(d_j) = +1$. Then, $v_{h_i(d_i)} = +1 \cdot y_1$, $v_{h_i(d_j)} = -1 \cdot y_2$, and $\omega_{h_i(d_i)} = -1 \cdot y_3$ and $\omega_{h_i(d_j)} = +1 \cdot y_4$. If $\omega_{h_i(d_i)} = +1$, then $y_1 = +1$, $y_2 = -1$, $y_3 = -1$, and $y_4 = +1$. Therefore, according to Eq. 6, we have

\[
e^{-2\epsilon} \leq \frac{\Pr[\nu_{h_i(d_i)} = \Theta_{h_i(d_i)}]}{\Pr[\omega_{h_i(d_i)} = \Theta_{h_i(d_i)}]} \cdot \frac{\Pr[\nu_{h_i(d_i)} = \Theta_{h_i(d_i)}]}{\Pr[\omega_{h_i(d_i)} = \Theta_{h_i(d_i)}]} \leq e^{2\epsilon}
\]

in which $\epsilon = \epsilon/2t$. Therefore, for each pair $(h_i, g_i)$ of hash functions in the above cases, it satisfies $\epsilon/t$-local differential privacy. Since there are $t$ hash functions to perturb the same symptom, $\mathcal{M}_{\text{Client-CS-LDP}}$ of each user satisfies $\epsilon$-local differential privacy according to Lemma 1. Since records between different users are disjoint, it satisfies $\epsilon$-differential privacy for all perturbed records according to Lemma 2.

Theorem 6. Protocol CS-LDP satisfies unbiased estimation. For any value $j^* \in [m]$, the variance of $f(j^*)$ is \( \left( \frac{1}{2} \cdot c_i^2 - 1 \right) \frac{f^2(j^*)}{f} \). eccentricity.

Proof. Protocol CS-LDP satisfies unbiased estimation if and only if $E[f(d)] = f(d)$. Assume that there is a fixed constant $j^* \in [m]$. For any $j \in [m]$, $Y_j$ is denoted as follows:

\[
Y_j = \begin{cases} 1 & \text{if } h(j) = h(j^*) \\ 0 & \text{otherwise} \end{cases}
\]

Meanwhile, assume that there is a perturbation function $p(\cdot)$, namely, Eq. 6. Then, $f(j^*)$ is equal to

\[
g(j^*) \cdot \mathbb{E}[h^*(j)]
= g(j^*) \cdot \sum_{j=1}^{n} g(d_j) p_c Y_j
= g(j^*) \cdot \sum_{j=1}^{n} f(j) g(j) p_c Y_j
= g(j^*) f(j^*) p_c Y_j + \sum_{j \neq j^*} f(d_j) g(d_j) g(d_j) p_c Y_j
\]

\[
145
\]
For the expected value of $k$ functions, we have

$$
\mathbb{E}[f(j)] = \mathbb{E}\left[f(j) \cdot p_c \cdot \sum_{j \neq j'} f(j') g(j') g(j) \cdot p_c \cdot Y_j \right]
$$

For $E[p]$, it is equal to $\frac{e^d - 1}{e^d + 1} \cdot \frac{1}{e^d + 1} = \frac{e^d - 1}{e^d + 1}$. Since hash functions $g$, are independent with each other, we have $E[g(j)g(j')Y_j] = E[g(j)]E[g(j')] = 0$ for $j \neq j'$. Therefore, $E[f(j')] = f(j')$.

For the variance of $f(j')$, $\text{Var}[f(j')]$ is derived as follows:

$$
\text{Var}[f(j')] = \mathbb{E}\left[ (f(j') - E[f(j')])^2 \right] = \mathbb{E}\left[ f(j')^2 \right] - (E[f(j')])^2
$$

$$
= \mathbb{E}\left[ (f(j')^2 - f(j')^2) \right] = \mathbb{E}\left[ f(j')^2 \right] - (E[f(j')])^2
$$

$$
= \mathbb{E}\left[ (f(j')^2 - 1) \cdot p_c \cdot g(j) g(j') \cdot p_c \cdot Y_j \right] + \mathbb{E}\left[ \left( \sum_{j \neq j'} f(j') g(j') g(j) \cdot p_c \cdot Y_j \right)^2 \right]
$$

$$
= \left( \sum_{j \neq j'} f(j') g(j') g(j) \cdot p_c \cdot Y_j \right)^2
$$

$$
= \left( \sum_{j \neq j'} f(j') g(j') g(j) \cdot p_c \cdot Y_j \right)^2
$$

$$
= \left( \sum_{j \neq j'} f(j') g(j') g(j) \cdot p_c \cdot Y_j \right)^2
$$

The first method for $k$-universal hash functions is implemented by the following equation:

$$
h(x) = \sum_{i=0}^{k-1} a_i x^i \mod p,
$$

where $p$ is a prime that is greater than value $x$ and $a_i$ is picked randomly from $[p]$. This equation was proposed by Wegman et al. [44]. It is obvious to see that this method is very simple and easy to implement. However, it has a disadvantage, especially when $p$ is very large. That is, if $p$ is an arbitrary prime, this method is fairly slow because the 'mod $p$' is relatively slow. In order to solve the problem, Thorup et al. [31] proposed a simple method, namely CW-Trick, in which $p$ is a so-called Mersenne prime of the form $2^r - 1$. In our experiments, we will use $p = 2^{61} - 1$. The CW-Trick hashing is used to $k = 2$.

6.3. Tabulation based hashing

If CW-Trick hashing is used to $k = 4$, the time is slow. Thorup et al. [31] proposed another method, namely Tabulation based hashing, to implement $k$-universal hash functions. They compared their method with CW-trick in terms of running time, which shows that the former is faster than the latter by at least a factor of 5. However, the weakness of Tabulation based hashing is that it requires large pre-computed tables and thus needs the extra space to store.

We firstly introduce Theorem 2.3 in [31] to compute hash value of $x$ by 4-universal hash functions.

**Theorem 7 (Theorem 2.3 in [31]).** There are any $q$ characters $\bar{x} = (x_0, x_1, \ldots, x_{q-1}), x_i \in \{2^r\}$. A $q \times r$ generator matrix, which satisfies that any square sub-matrix has full rank over prime field $\mathbb{Z}_p$, where $p \geq \max(2^r, q + r)$ is an odd prime. $\bar{y} = \bar{x}G$ are $r$ additional characters $(y_0, y_1, \ldots, y_{r-1})$. Then, hash function of $\bar{x}$ is

$$
h(\bar{x}) = h_0(x_0) \oplus \cdots \oplus h_{q-1}(x_{q-1}) \oplus h_0(y_0) \oplus \cdots \oplus h_{r-1}(y_{r-1}).
$$

$h(\cdot)$ is a 4-universal hashing function if $h_i$ and $h_j$ are independent 4-universal hash functions into $\{2^r\}$.

In our experiments, assume that value $d_i$ of each user $u_i$ has 32 bits and each character has 16 bits. Therefore, $q = 2$ and $r = 1$. In detail, value $d_i$ can be divided into two sub-values with 16-bits, i.e., $d_{i1}$ and $d_{i2}$ as the following equation:

$$
d_i(32 \text{ bits}) \rightarrow \begin{cases} 1010 \cdots 1010 \cdot 0101 \cdot \begin{array}{c} d_{i1}(16 \text{ bits}) \end{array} \cdot \begin{array}{c} d_{i2}(16 \text{ bits}) \end{array} \end{cases}
$$

1 Here, the length of $d_i$ is not constant. If there are a large number of elements in $D$, the length can be 64 or more bits and thus $q = 4$ and $r = 3$. 

6. Implementation of universal hash functions

In the section, we introduce the implementation of $k$-universal hash functions. In detail, we propose two different methods to implement universal hash functions. That is, CW-Trick hashing is used to the case in which $k = 2$ while Tabulation based hashing is used to $k = 4$. 

6.1. Universal hash function

In the protocols proposed by Section 4, the most important component is $k$-universal hash functions, which are defined as follows:

**Definition 5 (k-universal Class of Hash Functions [44]).** A class $\mathcal{H}$ of hash functions from $D$ into $[b]$ is a $k$-universal class of hash functions if for any distinct $d_0, d_1, \ldots, d_{k-1} \in D$ and any possibly identical $v_1, v_2, \ldots, v_{k-1} \in [b], 

$$
\forall i \in [k] \quad \Pr_{h \in \mathcal{H}}[h(d_i) = v_i] = 1/b^k,
$$

where $k$ is an positive integer.
Besides, 4-universal hash functions $h_i$ and $\bar{h}_j$ are implemented by tabulation based hashing. That is, we use precomputed tables to replace hash functions $h_i$ and $\bar{h}_j$. That is, Eq. (8) is changed as

$$h(\vec{x}) = T_0[x_0] \oplus \cdots \oplus T_{q-1}[x_{q-1}] \oplus T_0[y_0] \oplus \cdots \oplus T_{q+r-1}[y_{r-1}],$$

where $T_i$ is a fully random table of size $2^c$ with values from $\mathbb{R}$. Tabulation based hashing thus takes time $O(q+r)$ and space $O((q+r)2^c)$.

7. Experimental evaluation

In the section, our main goal is to study (1) the impact of different parameters on data utility; and (2) the application of protocols for frequency estimation of physical symptoms. The experiments are performed over two real datasets, representing two different distributions of symptoms.

7.1. Experimental setting

The evaluation is performed on a desktop computer, which has 4G of RAM and Inter(R) Pentium(R) CPU P6200 running Windows 7 operating system. All of protocols are implemented by Python. All of testing datasets are real.

Parameter settings. The number of hash function $\langle h_i, g_i \rangle$ is 100 and we will choose $t$ hash function pairs from them. The implementation of hash functions is based on CW-Trick and Tabulation-based hashing in Section 6. The values of $t$ and $b$ depend on error $\xi$ and confidence probability $\delta$. We mainly evaluate four protocols on different values of privacy budget $\epsilon$, error $\xi$ and confidence probability $\delta$.

Datasets. We use two real datasets to evaluate our protocols, including WISDM and BMBD. We select an attribute to estimate its frequency in these two datasets, which represent two different distributions of physical symptoms. The datasets are presented as follows:

- **WISDM** [45]. The dataset is collected from the accelerometer and gyroscope sensors of a smart-phone and smart-watch as 51 subjects. It is about diverse activities of daily living and has 15,630,426 records. We select attribute **Activity** from seven attributes to estimate its frequency. There are 18 activities, including walking, jogging, Sitting and so on.
- **BMBD** [46]. The dataset is collected from a public website Bixi, which offers bike rental service. It contains bike share information for Bixi Montreal from 2016 to 2019, including start and end station ID of a trip. We select attribute **Start_station_code** from six attributes to estimate its frequency.

Table 1 shows the statistical information of attribute **Activity** in WISDM and **Start_station_code** in BMBD. The distributions of these two attributes are different. **Activity** is uniformly distributed, while **Start_station_code** is loose. Therefore, we use them to evaluate our protocols for different distributions of symptoms.

### Table 1

| Attribute       | WISDM | BMBD |
|-----------------|-------|------|
| **Activity**    |       |      |
| Num.            | 18    | 621  |
| Min. ($f$)      | 833, 208 | 28 |
| Max. ($f$)      | 901, 381 | 188, 306 |
| Avg. ($f$)      | 868, 357 | 31, 501 |
| Med. ($f$)      | 870, 532 | 25, 238 |
| Total           | 15, 630, 426 | 19, 561, 549 |

**Fig. 3.** Results for Top-18 of WISDM.
The concrete metrics are presented as follows:

- **Relative Error (RE).** The metric is usually used for SUM/COUNT/AVG queries and measures how large the error is relatively to the true answer for each query \(f(i)\) by the following equation:

\[
RE = \frac{1}{h} \sum_{j=0}^{h-1} \left| \frac{f(i) - f'(i)}{f(i)} \right|, \quad i_j \in [m] \tag{10}
\]

Furthermore, for the top-\(k\) elements, we evaluate them by the average and median value of relative errors, i.e., ARE and MRE.

- **Mean Square Error (MSE).** It is used to measure the estimation accuracy by the average of the squared errors, namely,

\[
MSE = \frac{1}{h} \sum_{j=0}^{h-1} \left( f(i) - f'(i) \right)^2, \quad i_j \in [m] \tag{11}
\]

where \(f(i)\) is the true frequency of users taking value \(i_j\).

### 7.2. Experimental results

In the following section, we show the impact of three key parameter \(\xi, \delta, \text{and} \epsilon\) in terms of RE and MSE for different datasets. Then, we show the effectiveness of protocols by estimating top-\(k\) in different cases. In order to reduce the error, we repeat each experiment 10 times. Since frequency of different records in BMBD is different, we will consider two cases, including top-10 and top-30. That is, we estimate 10 and 30 elements with the most frequency, respectively.

#### 7.2.1. Impact of parameter \(\xi\)

In order to analyze impact of parameter \(\xi\), we firstly set the parameters. That is, confidence probability \(\delta\) is 0.1 and privacy budget \(\epsilon\) is 3. \(\xi\) of WISDM is \([0.15, 0.18, 0.2, 0.25]\) while that is \([0.004, 0.0044, 0.005, 0.00571]\) for BMBD.

Figs. 3(a)–3(d) show RE and MSE of WISDM by different protocols, including CMS-LDP, FCS-LDP, CS-LDP, and FAS-LDP. It is intuitive to see that RE and MSE in CMS-LDP and FCS-LDP increase as \(\xi\) increases from 0.15 to 0.25 in Figs. 3(a) and 3(c). This is consistent to CMS and FCS without LDP. This implies that addition of privacy preservation does not change influence of parameter \(\xi\). However, Figs. 3(b) and 3(d) show that \(\xi\) has little impact on CS-LDP and FAS-LDP, since RE and MSE do not change as \(\xi\) increases from 0.15 to 0.25.

Figs. 4(a)–4(d) and 5(a)–5(d) show RE and MSE of BMBD in two different cases by four private protocols. Compared to WISDM, it has the opposite results for protocol CMS-LDP and FCS-LDP. That is, with \(\xi\) increasing from 0.004 to 0.00571, RE and MSE decrease. The key factor for this opposition is size of \(b\) mapped by hash function \(H\), in which the size in BMBD is far greater than that in WISDM. Thus, perturbation for privacy preservation is the dominant factor for BMBD, while conflict probability is relatively less. As size of \(b\) decreases, it leads to less expected error. Figs. 4(b) and 4(d) and Figs. 5(b) and 5(d) show that RE and MSE of BMBD by CS-LDP and FAS-LDP have no much difference with \(\xi\) increasing. That is, \(\xi\) has little impact on CS-LDP and FAS-LDP.

Therefore, the impact of \(\xi\) on CMS-LDP and FCS-LDP depends on the size of \(b\). In contrast, it has little impact on CS-LDP and FAS-LDP.

#### 7.2.2. Impact of parameter \(\delta\)

Here, we consider the impact of parameter \(\delta\) for four LDP protocols. The expected error \(\xi\) is 0.18 for WISDM, while that is 0.005 for BMBD. Privacy budget \(\epsilon\) is 3. The range of \(\delta\) is \([0.005, 0.02, 0.1, 0.25]\).
For WISDM, Figs. 3(e)–3(h) show RE and MSE of four protocols. It is intuitive to find that RE and MSE increase with $\delta$ increasing. This implies that adding the number of hash functions can efficiently improve the accuracy for perturbed data. Meanwhile, it needs more time for the users and the third party. From Figs. 3(e)–3(h), RE and MSE are fast increasing once $\delta$ is greater than 0.25. Therefore, $\delta < 0.25$ is a good choice.

For BMBD, Figs. 4(e)–4(h) and 5(e)–5(h) show RE and MSE of four protocols. Two cases have the similar results. It is easy to see that the impact of $\delta$ on CMS-LDP and FCS-LDP is counter to that on CS-LDP and FAS-LDP. In detail, RE and MSE of CMS-LDP and FCS-LDP decrease as $\delta$ increases while those of CS-LDP and FAS-LDP have the opposite results. This means that it should add the number of hash functions for FCS-LDP and FCS-LDP and decrease the number for CS-LDP and FAS-LDP.

Thus, the impact of $\delta$ on CS-LDP and FAS-LDP is consistent, while that on CMS-LDP and FCS-LDP is changing in different datasets.

### 7.2.3. Impact of parameter $\epsilon$

In order to analyze the impact of $\epsilon$, $\xi$ for WISDM is 0.18, while that is 0.005 for BMBD. Confidence probability $\delta$ is 0.1. The range of privacy budget $\epsilon$ is $[1, 3, 5, 7]$.

For WISDM, Figs. 3(i)–3(l) show RE and MSE of four protocols. Figs. 3(i) and 3(k) show that RE and MSE of CMS-LDP and FCS-LDP do not decrease with $\epsilon$ increasing. In contrast, they have big fluctuation. Compared to those for BMBD in Figs. 4(i), 4(k), 5(i) and 5(k), the possible reason is the size of $b$. For CS-LDP and FAS-LDP, the accuracy increases as privacy budget increases. However, it does not change once privacy budget exceeds some threshold (e.g., 5 in Figs. 3(b) and 3(d)).

For BMBD, Figs. 4(i)–4(l) and 5(i)–5(l) show RE and MSE of four protocols. We can find that top-10 and top-30 have the same results about RE and MSE. For four protocols, both of RE and MSE decrease as privacy budget $\epsilon$ increases. In other words, the increase of RE and MSE in CMS-LDP and FCS-LDP is fast, while that in FCS-LDP and FAS-LDP is not obvious. Once privacy budget exceeds some threshold (e.g., 3 in Fig. 4(i)), RE and MSE have little change for CS-LDP and FAS-LDP. The threshold is 5 for CMS-LDP and FCS-LDP.

According to the above results, we can derive that privacy budget $\epsilon$ has the important impact on four protocols. Meanwhile, once $\epsilon$ exceeds some threshold, the impact can be ignored.

### 7.2.4. Protocol applications in IoMT

By our experiments, we analyze two different distributions of symptoms for some infectious disease in IoMT. That is, the symptoms are uniformly distributed in case (i), while those are loose in case (ii).

Generally speaking, we have two important findings in Figs. 3–5. The first one is that the utility of CMS-LDP and FCS-LDP is greater than that of CS-LDP and FAS-LDP. Since the time of CMS-LDP and FCS-LDP is smaller than that of CMS-LDP and FAS-LDP, Therefore, CMS-LDP and FCS-LDP are relatively optimal in terms of utility and computation overhead. The second one is that protocol CMS-LDP is not absolutely better than FCS-LDP, and vice versa. Furthermore, different distributions of symptoms have an important influence for the choices of different parameters, including $\xi$, $\delta$ and $\sigma$. In case (i), FCS-LDP has better stability than CMS-LDP as different parameters change. In case (ii), CMS-LDP and FCS-LDP have the similar changes. Parameter $\xi$, $\delta$ and $\sigma$ should be as small as possible to maximize the utility.

### 8. Conclusions

The focus of the paper is on how to estimate the frequency of symptoms under local differential privacy for infectious disease analysis.
in IoMT. Based on basic sketches, this paper has proposed four protocols, including CMS-LDP, FCS-LDP, CS-LDP and FAS-LDP. We have proved that our proposed protocols satisfy two important properties, i.e., unbiased estimation and local differential privacy. Meanwhile, we have presented the variance of four protocols. Through empirical analysis, CMS-LDP and CS-LDP are relatively optimal protocols for frequency estimation of physical symptoms in IoMT. We plan to design better encoding methods to reduce the computation complexity and perturbation techniques to improve the accuracy of our protocols in the future.

CRediT authorship contribution statement

Xiaotong Wu: Conceptualization, Methodology, Formal analysis, Writing - original draft. Mohammad Reza Khosravi: Conceptualization, Methodology, Writing - original draft. Lianyong Qi: Conceptualization, Writing - original draft. Wanchun Dou: Supervision, Writing - review & editing, Project administration. Xiaolong Xu: Conceptualization, Writing - review & editing, Investigation, Validation, Supervision.

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

Our thanks to the reviewers for their constructive comments and suggestions to improve the quality of the manuscript. This work was supported in part by the National Natural Science Foundation of China under Grant No. 41971343.

References

[1] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, L. Qi, Trust-oriented iot service placement for smart cities in edge computing, IEEE Internet Things J. (2019).
[2] Y. Zhang, C. Yin, Q. Wu, Q. He, H. Zhu, Location-aware deep collaborative filtering for service recommendation, IEEE Trans. Syst. Man Cybern.: Syst. (2019) 1–12.
[3] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, Q. He, Efficient query of quality correlation for service composition, IEEE Trans. Serv. Comput. (2018) 1.
[4] F. Wu, S. Zhao, B. Yu, Y.-M. Chen, W. Wang, Z.-G. Song, Y. Hu, Z.-W. Tao, J.-H. Tian, Y.-Y. Pei, M.-L. Yuan, Y.-L. Zhang, F.-H. Dai, Y. Liu, Q.-M. Wang, J.-J. Zheng, L. Xu, E.C. Holmes, Y.-Z. Zhang, A new coronavirus associated with human respiratory disease in China, Nature 579 (2020) 265–269.
[5] S. Deep, X. Zheng, A. Jolfaei, D. Yu, P. Ostovari, A. K. Bashir, A survey of security and privacy issues in the internet of things from the layered context, Trans. Emerg. Telecommun. Technol. (2020).
[6] G. Harizanis, O. Souhatos, S. Ioannidis, C. Verikoukis, G. Demetriou, C. Tsalousis, Review of security and privacy for the Internet of Medical Things (IoMT), in: Proceedings of 15th International Conference on Distributed Computing in Sensor Systems, DCOSS, 2019, pp. 457–464.
[7] T. Wang, Z. Zheng, A. Bashir, A. Jolfaei, Y. Xu, Fingerprinting: A data protection and privacy preserving mechanism for fingerprint identification, ACM Trans. Internet Technol. (2020).
[8] S. Mirzamohammadi, A.A. Sani, Viola: Trustworthy sensor notifications for enhanced privacy on mobile systems, IEEE Trans. Mobile Comput. 17 (11) (2018) 2689–2702.
[9] L. Qi, X. Wang, X. Xu, W. Dou, S. Li, Privacy-aware cross-platform service recommendation based on enhanced locality-sensitive hashing, IEEE Trans. Netw. Sci. Eng. (2020).
[10] X. Xu, Q. Liu, X. Zhang, J. Zhang, L. Qi, W. Dou, A blockchain-powered crowdsourcing method with privacy preservation in mobile environment, IEEE Trans. Comput. Soc. Syst. 6 (6) (2019) 1407–1419.
[11] C. Zuo, L. Yin, Y. Zhang, Why does your data leak? Uncovering the data leakage in cloud from mobile apps, in: Proceedings of Symposium on Security and Privacy, SP, IEEE, 2019, pp. 1296–1310.
[12] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, M.Z.A. Bhuiyan, Joint optimization of offloading utility and privacy for edge computing enabled iot, IEEE Internet Things J. (2019).
[13] C. Dwork, Differential privacy: A survey of results, in: Proceedings of International Conference on Theory and Applications of Models of Computation, TAMC, Vol. 4978, Springer, 2008, pp. 1–19.
[14] Z. Zheng, T. Wang, J. Wen, S. Mumtaz, A.K. Bashir, S.H. Chaudhary, Differentially private high-dimensional data publication in internet of things, IEEE Internet Things J. 7 (4) (2020) 2640–2650.
[15] J.C. Duchi, M.I. Jordan, M.I. Wainwright, Local privacy and statistical minimax rates, in: Proceedings of 54th Annual Symposium on Foundations of Computer Science, FOCS, IEEE, 2013, pp. 429–438.
[16] G. Cormode, T. Kulkarni, D. Srivastava, Marginal release under local differential privacy, in: Proceedings of the International Conference on Management of Data, SIGMOD, ACM, 2018, pp. 131–146.
[17] J.C. Duchi, M.I. Wainwright, M.I. Jordan, Minimax optimal procedures for locally private estimation, J. Amer. Statist. Assoc. 113 (521) (2018) 182–201.
[18] N. Wang, X. Xiao, Y. Yang, Z. Zhao, S.C. Hui, H. Shin, J. Shin, G. Yu, Collecting and analyzing multidimensional data with local differential privacy, in: Proceedings of International Conference on Data Engineering, ICDE, IEEE, 2019, pp. 638–649.
[19] M. Ye, A. Barg, Optimal schemes for discrete distribution estimation under locally differential privacy, IEEE Trans. Inform. Theory 64 (8) (2018) 5662–5676.
[20] R. Bassily, K. Nissim, U. Stemmer, A.G. Thakurta, Practical locally private heavy hitters, in: Proceedings of Annual Conference on Neural Information Processing Systems, 2017, pp. 2288–2296.
[21] R. Bassily, A.D. Smith, Local, private, efficient protocols for succinct histograms, in: Proceedings of the Forty-Seventh Annual Symposium on Theory of Computing, STOC, ACM, 2015, pp. 127–135.
[22] M. Bun, J. Nelson, U. Stemmer, Heavy hitters and the structure of local privacy, ACM Trans. Algorithms 15 (4) (2019) 51:1–51:40.
[23] Z. Qin, Y. Yang, T. Yu, L. Khali, X. Xiao, K. Ren, Heavy hitter estimation over sensitive data with local differential privacy, in: Proceedings of the Conference on Computer and Communications Security, CCS, ACM, 2016, pp. 192–203.
[24] T. Wang, J. Blocki, N. Li, S. Jha, Locally differentially private protocols for frequency estimation, in: Proceedings of USENIX Security Symposium, 2017, pp. 729–745.
[25] S.L. Warner, Randomized response: A survey technique for eliminating evasive answer bias, J. Amer. Statist. Assoc. 60 (309) (1965) 63–69.
[26] S. Wang, L. Huang, P. Wang, H. Deng, H. Xu, W. Yang, Private weighted histogram aggregation in crowdsourcing, in: Proceedings of International Conference on Wireless Algorithms, Systems, and Applications, WASA, Vol. 9798, Springer, 2016, pp. 295–307.
[27] Ú. Erlingson, V. Pihur, A. Korolova, RAPPOR: randomized aggregatable privacy-preserving ordinal response, in: Proceedings of the Conference on Computer and Communications Security, CCS, ACM, 2014, pp. 1054–1067.
[28] M. Charikar, K.C. Chen, M. Farach-Colton, Finding frequent items in data streams, in: Proceedings of International Colloquium on Automata, Languages and Programming, ICALP, 2002, pp. 693–703.
[29] G. Cormode, S. Muthukrishnan, An improved data stream summary: The count-min sketch and its applications, J. Algorithms 55 (1) (2005) 58–75.
[30] G. Cormode, M.N. Garofalakis, Sketching streams through the net: Distributed approximate aggregate tracking, in: Proceedings of the 31st International Conference on Very Large Data Bases, VLDB, 2005, pp. 13–24.
[31] M. Thorup, Y. Zhang, Tabulation based 4-universal hashing with applications to second moment estimation, in: Proceedings of the Fifteenth Annual Symposium on Discrete Algorithms, SODA, 2004, pp. 615–624.
[32] L. Sweeney, K-anonymity: A model for protecting privacy, Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 10 (5) (2002) 557–570.
[33] M. Machanavajjhala, D. Kifer, J. Gehrke, M. Venkitasubramaniam, L-diversity: Privacy beyond k-anonymity, ACM Trans. Knowl. Discovery Data 1 (1) (2007) 1.
[34] D. McSherry, F. McSherry, K. Nissim, A.D. Smith, Calibrating noise to sensitivity in private data analysis, in: Proceedings of Third Theory of Cryptography Conference, TCC, Vol. 3876, Springer, 2006, pp. 265–284.
[35] F. McSherry, K. Talwar, Mechanism design via differential privacy, in: Proceedings of 48th Annual IEEE Symposium on Foundations of Computer Science, FOCS, IEEE, 2007, pp. 94–103.
[36] P. Kairouz, S. Oh, P. Viswanath, Extremal mechanisms for local differential privacy, in: Proceedings of Advances in Neural Information Processing Systems, 2014, pp. 2879–2887.
[37] P. Kairouz, S. Oh, P. Viswanath, Extremal mechanisms for local differential privacy, in: Proceedings of Symposium on Security and Privacy, SP, IEEE, 2019, pp. 317–331.
[38] X. Ye, H. Hu, X. Meng, H. Zheng, Privkv: Key-value data collection with local differential privacy, in: Proceedings of Symposium on Security and Privacy, SP, IEEE, 2019, pp. 317–331.
[42] R. Chen, H. Li, A.K. Qin, S.P. Kasiviswanathan, H. Jin, Private spatial data aggregation in the local setting, in: Proceedings of IEEE 32nd International Conference on Data Engineering (ICDE), 2016, pp. 289–300.

[43] F. McSherry, Privacy integrated queries: an extensible platform for privacy-preserving data analysis, in: Proceedings of International Conference on Management of Data, SIGMOD, ACM, 2009, pp. 19–30.

[44] M.N. Wegman, L. Carter, New hash functions and their use in authentication and set equality, J. Comput. System Sci. 22 (3) (1981) 265–279.

[45] G. Weiss, WISDM smartphone and smartwatch activity and biometrics dataset data set. http://archive.ics.uci.edu/ml/index.php.

[46] J. Wang, Bixi montreal bikeshare data - Bikeshare information for bixi montreal. https://www.kaggle.com/jackywang529/bixi-montreal-bikeshare-data.

[47] Y. Zhang, K. Wang, Q. He, F. Chen, S. Deng, Z. Zheng, Y. Yang, Covering-based web service quality prediction via neighborhood-aware matrix factorization, IEEE Trans. Serv. Comput. (2019) 1.

[48] X. Xue, H. Han, S. Wang, C. Qin, Computational experiment-based evaluation on context-aware O2O service recommendation, IEEE Trans. Serv. Comput. 12 (6) (2019) 910–924.

[49] X. Xue, S. Wang, L. Zhang, Z. Feng, Y. Guo, Social learning evolution (SLE): computational experiment-based modeling framework of social manufacturing, IEEE Trans. Ind. Inform. 15 (6) (2019) 3343–3355.