A New Lower Bound of Privacy Budget for Distributed Differential Privacy

Zhigang Lu*, Hong Shen*†
*School of Computer Science, The University of Adelaide, Adelaide, Australia
Email: {zhigang.lu, hong.shen}@adelaide.edu.au
†School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China

***Abstract*—Distributed data aggregation via summation (counting) helped us to learn the insights behind the raw data. However, such computing suffered from a high privacy risk of malicious collusion attacks. That is, the colluding adversaries infer a victim's privacy from the gaps between the aggregation outputs and their source data. Among the solutions against such collusion attacks, Distributed Differential Privacy (DDP) shows a significant effect of privacy preservation. Specifically, a DDP scheme guarantees the global differential privacy (the presence or absence of any data curator barely impacts the aggregation outputs) by ensuring local differential privacy at the end of each data curator. To guarantee an overall privacy performance of a distributed data aggregation system against malicious collusion attacks, part of the existing work on such DDP scheme aim to provide an estimated lower bound of privacy budget for the global differential privacy. However, there are two main problems: low data utility from using a large global function sensitivity; unknown privacy guarantee when the aggregation sensitivity of the whole system is less than the sum of the data curator's aggregation sensitivity. To address these problems while ensuring distributed differential privacy, we provide a new lower bound of privacy budget, which works with an unconditional aggregation sensitivity of the whole distributed system. Moreover, we study the performance of our privacy bound in different scenarios of data updates. Both theoretical and experimental evaluations show that our privacy bound offers better global privacy performance than the existing work.**

***Keywords*-distributed computing; differential privacy; privacy budget**

I. Introduction

Because of the prevalence of smart devices, such as smartphones, fitness wristband and wireless sensors, personal information is generated at every moment everywhere from these distributed devices. Among all the functions in distributed data analysis, summation and counting function are two of the most popular ones. For example, the energy providers optimise electricity allocation in a state based on the periodic power usage (summation) in suburbs of the state; a government health department learns the seasonal flu trends by counting the daily number of flu victims from different local hospitals.

However, malicious collusion attacks brought serious privacy disclosure risks on the above applications. In general, the colluding malicious adversaries infer a victim's private data from the gaps between the aggregation outputs and the source data of adversaries. Figure 1 shows the basic settings of network topology and assumptions in this paper. Specifically, there are n data curators send their local answer to a query f to a data aggregator for the final output of the distributed system. We assume the communications between data curators and aggregator are secure; the data aggregator and at most $n-1$ out of n data curators are maliciously colluding. Therefore, in this paper, we focus on the privacy issues in summation (counting) function in a distributed scenario against the worst case malicious collusion attacks.

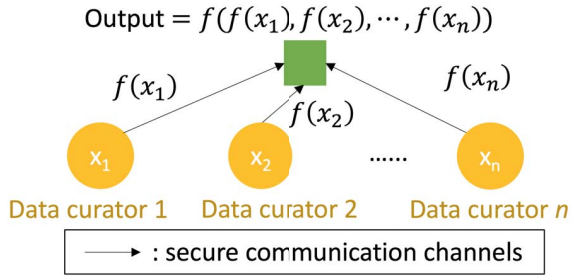


Figure 1: Basic Settings of Network Topology.

To preserve privacy in distributed summation (counting), several techniques have been proposed, such as Secure Multiparty Computation (SMC or MPC) paradigm [1], Anonymisation [2] and Differential Privacy [3]. However, due to the vulnerability to malicious collusion attacks (analysis in Section II), the family of SMC and the anonymity-based techniques are not suitable to preserve privacy under our scenario assumption. While differential privacy shows a better privacy preserving performance in information theory in the past decade. The traditional differential privacy, usually, works in centralised scenarios where a trusted and secure database keeps all the data. However, since in our assumptions, the data are stored in distributed servers, we need apply the extended concepts of Distributed Differential Privacy (DDP) in this paper.

In the research field of DDP, a group of work, inspired by Dwork et al. [4], extended the traditional differential privacy into distributed scenarios by combining the techniques from SMC and differential privacy in different applications. However, the techniques of SMC in these works will contribute a significant computational cost which is not realistic to the smart devices. Differing from the above, McSherry [5] and Shi et al. [6] gave a more formal definition of DDP. That

0-7695-6330-9/17/31.00 ©2017 IEEE
DOI 10.1109/PDCAT.2017.00014

is, a DDP scheme guarantees the global differential privacy (the presence or absence of any data curator barely impacts the aggregation outputs) automatically by ensuring local differential privacy at the end of each data curator. Despite the low time complexity in [5] and [6], the current two lower bounds of the global privacy budget to achieve such DDP suffered from two main problems: low data utility by using aggregator's global function sensitivity (defined in Equation (2)) at the end of each data curator in [6][1]; unknown privacy guarantee when the aggregator's local function sensitivity ($\Delta f_L(X)$, defined in Equation (1)) is less than the sum of the data curator's local function sensitivity ($\sum_i \Delta f_L(x_i)$) in [5].

Problem statement. In a distributed data aggregation system, a data curator i, saves time serial data, $x_{i,j}$, at time j. For a summation (or counting) function f, the local function sensitivity at the end of data aggregator (Equation (1)) is $\Delta f_L(X) = \max_{\forall \text{neighbouring} X'} |f(X) - f(X')| = \max_i\{f(x_i)\} = \max_i\{\sum_j x_{i,j}\}$, where $X = \{x_1, x_2, \cdots, x_n\}$; the local function sensitivity at the end of data curators is $\Delta f_L(x_i) = \max_j\{x_{i,j}\}$. We may have a case that some of the data curators always return large values, e.g., a smart meter at a factory always reads greater electricity consumption than a smart meter at a residential apartment at any time. Then, in such case, we have: $\Delta f_L(X) = \max_i\{f(x_i)\} = \max_i\{\sum_j x_{i,j}\} \geq \sum_i\{\max_j x_{i,j}\} = \sum_i \Delta f_L(x_i)$. Namely, $\Delta f_L(X) \geq \sum_i \Delta f_L(x_i)$ which is the condition of the lower bound of privacy budget in [5]. However, in some other cases (e.g., the MovieLens dataset used in our experiments), the above condition will not be achieved. Table I shows an example on summation function that $\Delta f_L(X) < \sum_i \Delta f_L(x_i)$, where $\Delta f_L(X) = \max\{10, 26, 30\} = 30$, $\sum_i \Delta f_L(x_i) = 4 + 8 + 27 = 39$. Therefore, in this paper, we aim to provide a new lower bound of privacy budget on summation (counting) function without condition of functions sensitivity.

Table I: A counterexample to [5].

j	x_1	x_2	x_3
1	1	5	1
2	2	6	1
3	3	7	1
4	4	8	27
$f(x_i)$	10	26	30
$\Delta f_L(x_i)$	4	8	27

Contributions. In this paper, to address the unknown privacy guarantee problem in [5], we provide an unconditional lower bound of privacy budget to guarantee distributed differential privacy for summation (counting) function against the strongest collusion attacks ($n-1$ out of n data curators are colluders). The main contributions of this paper are listed below:

- Our lower bound of privacy budget works with the local function sensitivity at the end of data aggregator (much smaller than the global function sensitivity) which provides better data utility than the naive scheme in [6].
- The local function sensitivity at the end of data aggregator used in this paper does not rely on any condition. That is, our differential privacy guarantee (lower bound of privacy budget) for distributed summation (counting) function is more adaptable in general cases than [5].
- We provide a theoretical evaluation of the privacy performance of our lower bound of privacy budget in two scenarios: passive data updates (malicious data cleansing) and active data updates (dynamically adding or removing data curator, serially refreshing data).
- We provide an experimental evaluation to analyse the performance of our privacy budget bound over various experimental settings.

Organisation. We organise the rest of the paper as follows. In Section II, we briefly summarise and analyse the existing schemes to achieve distributed differential privacy on summation (counting) function. Afterwards, in Section III we show the adversary model in this paper, then formally introduce the definitions of differential privacy and distributed differential privacy. Next, we show how to compute our new lower bound of privacy budget and the related proofs in Section IV. In Section V, we study two data updates scenarios in real-life. Then, in Section VI, we experimentally evaluate the performance of our privacy budget bound with two real-world datasets. Finally, Section VII concludes this paper.

II. Related Work

In this section, we will discuss the existing solutions against the strongest collusion attacks ($n-1$ out of n data curators are colluders) by applying differential privacy in a distributed scenario for both advantages and disadvantages. Particularly, part of the researchers [4], [7]–[10] in this field combined centralised differential privacy with Secure Multiparty Computation (SMC) paradigm in distributed environment. Another group of the researchers [5], [6], while, applied differential privacy directly in a distributed scenario and studied the overall privacy performance (decided by the privacy budget of differential privacy) of the distributed data aggregation system.

Roughly, the schemes, combined SMC and centralised differential privacy, have two steps: firstly, every data curator adds the differentially private noise on their data to guarantee the local differential privacy; secondly, all the data curators jointly compute the query function securely by a selected

[1]In this paper, we use the naive scheme of [6] because it is same as [5] and this paper that guarantee the global differential privacy only by adding local differentially private noises at the end of data curators.

method in SMC paradigm (we choose a particular method of SMC according to the computation requirements). Because SMC was born to enhance the security of computing in distributed computing and differential privacy provides the strongest assumption of an adversary's background knowledge to protect individual privacy, at first glance, such combination perfectly guarantees both the security of computing process and the privacy of computing outcomes for the distributed data aggregation system.

However, the combination schemes have two main problems: failed guarantee of differential privacy for the whole system (i.e. global differential privacy) and high computational. Particularly, the reason of failed guarantee of global differential privacy is that these methods did not differentiate local differential privacy and global differential privacy in the network of Figure 1. To provide a differential privacy guarantee, we need two parameters for noise generation: a privacy budget ϵ and the given query function's sensitivity Δf, where Δf is decided by a special atomic item in the input database or dataset (a formal definition of Δf is in Section III). That is, in the local ϵ-differential privacy, the atomic item is a record in a data curator's database (see y_i in Definition 2); while in the global ϵ-differential privacy, the atomic item will be a data curator's database (see $f(Y)$ in Definition 2). But, in the existing work, when the authors prove their scheme is differential privacy globally, a Δf from the data curator's local differential privacy was used instead of the Δf of the whole distributed system. Moreover, these schemes are not energy-efficient. Considering the time complexity of the techniques of SMC, it is not realistic to run such heavy program in the battery-powered sensors/phones.

While some researchers focus on analysing the privacy performance of global differential privacy via ensuring local differential privacy. McSherry [5] studied the property of differential privacy in parallel composition following the topology showed in Figure 1 (Theorem 4 in [5]). The property of parallel composition ensures that if each data curator i provides ϵ_i-differential privacy locally, and all data curator's local database is disjoint to each other, then a parallel composition of these data curators will have $\max_i\{\epsilon_i\}$-differential privacy globally. To be honest, this property is helpful for analysing the privacy performance of a distributed system, such as [11], [12]. However, such parallel composition property has introduced a condition in the proof, that is, $\Delta f_L(X) \geq \sum_i \Delta f_L(x_i)$, where $\Delta f_L(X)$ is the local function sensitivity for $\max\{\epsilon_i\}$-differential privacy at the end of aggregator; $\Delta f_L(x_i)$ is the local function sensitivity for ϵ_i-differential privacy at the end of curators; $X = (x_1, x_2, \ldots, x_n)$. Namely, when $\Delta f_L(X) < \sum_i \Delta f_L(x_i)$, the global differential privacy guarantee is unknown in [5]'s parallel composition.

Compared with [5], Shi et al. [6] achieved $\sum_i \epsilon_i$-differential privacy globally when each data curator provides ϵ_i-differential privacy locally without any conditions. However, in [6], both local and global differential privacy use a same function sensitivity $\Delta f = \Delta f_{Global}$ (Please note that the "Global" here means the global function sensitivity, defined as Equation (2), which is a term in differential privacy). Actually, the large global function sensitivity Δf_{Global} will introduce more noises which will cause a low data utility for the final aggregation outputs.

Therefore, based on the above analysis on the existing work to guarantee distributed differential privacy for summation (counting) function, in this paper, we aim to provide a lower bound of privacy budget with an unconditional system's local function sensitivity (in Equation (1)) to guarantee a better privacy performance.

III. Preliminaries

In this section, we will introduce the adversary model, definition of differential privacy, including centralised differential privacy and distributed differential privacy, we used in this paper briefly.

A. Adversary Model

Generally, we categorise the adversary into two models, semi-honest adversary and malicious adversary, according to an adversary's willingness to follow a privacy preserving protocol. In this paper, we assume the data aggregator and at most $n-1$ out of n data curators are malicious.

Malicious Adversary. A malicious adversary is also known as an active adversary who aims to cheat the protocol arbitrarily to disclose a targeted victim's privacy. The malicious adversary can work alone (e.g. pollution attack by providing random inputs to a system) or together (e.g. collusion attacks by cooperating with other colluded adversaries) to enhance their abilities. In this paper, we address the privacy issues against malicious collusion attacks.

B. Differential Privacy

Differential privacy is one of the most popular notion of privacy in the current research field of privacy preservation [13]. The concept of differential privacy was firstly introduced and defined by Dwork et at. as ϵ-indistinguishability [14], then formalised in [3]. Informally, differential privacy is a scheme that minimises the sensitivity of output for a given statistical query function (e.g. summation, counting) on two neighbouring (differentiated in one record) datasets. In general, differential privacy guarantees the presence or absence of any record in a database will be concealed to an adversary.

Centralised Differential Privacy. In centralised differential privacy, a central honest server guarantees the computing outputs are differentially private. The basic setting is a pair of neighbouring dataset X and X', where X', owned by an adversary, contains the information of all the entries except one record in the database X, owned by a trusted server. For

convenience of the proof in this paper, a formal definition of differential privacy by McSherry [5] is shown as follow:

Definition 1 (ϵ-Differential Privacy [5]): A randomised mechanism $\mathcal{T}$ is ϵ-differential privacy if for all neighbouring datasets X and X', and for all output sets $S \subseteq Range(T)$, $\mathcal{T}$ satisfies:

$$\frac{\Pr[\mathcal{T}(X) \in S]}{\Pr[\mathcal{T}(X') \in S]} \leq \exp(\epsilon \times \Delta f(X)),$$

where ϵ is the privacy budget, $\Delta f(X)$ is the sensitivity of a function f.

Normally, we have two types of $\Delta f(X)$, local sensitivity $f_L(X)$ [15] and global sensitivity $f_G(X)$ [14], where

$$\Delta f_L(X) = \max_{\forall X'} |f(X) - f(X')|, \tag{1}$$

$$\Delta f_G(X) = \max_{\forall X} \{\Delta f_L(X)\}. \tag{2}$$

The privacy budget ϵ is set by the data curator according to $\Delta f(X)$. To decrease the noise injected to the original data, we usually use $\Delta f_L(X)$ as the value for $\Delta f(X)$ [15].

In this paper, because we focus on the summation (counting), which is a numeric function, we use the Laplace mechanism [3] to achieve differential privacy guarantee by adding random noise with Laplace distribution in the output of computation.

Distributed Differential Privacy. Currently, there is no unified definition on Distributed Differential Privacy yet. Because we need to differentiate the local and global differential privacy (see Section II for reasons) when guaranteeing the differential privacy for a distributed system, in this paper, we use the definition of distributed differential privacy in [6].

Definition 2 (ϵ-Distributed Differential Privacy [6]): A parallel system consisting of n distributed data curators with database x_i (Figure 1) is ϵ-distributed differentially private, if for a query function f we have:

$$f(Y) = f(X) + DPnoise(\epsilon, \Delta f(X)), \tag{3}$$

where $X = (x_1, x_2, \ldots, x_n)$, $Y = (y_1, y_2, \ldots, y_n)$, $y_i = f(x_1) + DP(\epsilon_i, \Delta f(x_i))$, $DPnoise$ is the differentially private noise, ϵ_i is the local privacy budget of data curator i, ϵ is the global privacy budget, Δf is the sensitivity of query function f.

In Definition 2, we need to be aware that the function sensitivity at the end of data curator i, $\Delta f(x_i)$, and the function sensitivity at the end of data aggregator, $\Delta f(X)$, are computed by different inputs. Moreover, we achieve distributed differential privacy in Definition 2 by adding differentially private noise at the end of data curator only. That is, the left-hand side of Equation (3), $f(Y)$, is the aggregation of local differential privacy; while the right-hand side of Equation (3) is the measurement of overall privacy performance for the whole distributed data aggregation system.

IV. Achieving Distributed Differential Privacy against Collusion Attacks

In this paper, our major privacy risk comes from the worst collusion attacks where $n-1$ out of n data curators are colluding. Particularly, the colluders share the raw data (inputs), then attempt to infer a victim's privacy by comparing the published aggregation outputs and their shares.

To guarantee ϵ-distributed differential privacy to preserve data curator's privacy against the worst collusion attacks. Based on Definition 2, we should guarantee the global differential privacy (with local sensitivity) for the whole distributed system (i.e. at the end of aggregator) by ensuring local differential privacy at the end of each data curator.

According to Definition 1 and Definition 2, for a mechanism $\mathcal{T}$ to be ϵ-global differentially private, we should have:

$$\mathcal{T}(X) = f(X) + DPnoise(\epsilon, \Delta f(X)),$$

namely,

$$\frac{\Pr[\mathcal{T}(X) \in S]}{\Pr[\mathcal{T}(X') \in S]} \leq \exp(\epsilon \times \Delta f(X)), \tag{4}$$

where $X = (x_1, x_2, \ldots, x_n)$, X' is a neighbouring set of X, $|\|X\|_0 - \|X'\|_0| = 1$.

Note that the global differential privacy is the one which preserves privacy against the strongest collusion attacks. Then we have the following Lemma 1 to find a lower bound of privacy budget ϵ to achieve the global differential privacy in Equation (4), where the set of all data curators is $\mathcal{U}$.

Lemma 1: Let $\mathcal{T}_i$ provide ϵ_i-differential privacy, $\forall i \in \mathcal{U}$, then the parallel composition of $\mathcal{T}_i$ (as Figure 1) will be $\max_{i \in \mathcal{U}} \left\{ \frac{|\mathcal{U}| \cdot \Delta f_L(x_i)}{f(x_i)} \epsilon_i \right\}$-differentially private.

Proof: For each data curator i, we have guaranteed the local differential privacy,

$$\mathcal{T}_i(x_i) = f(x_i) + DPnoise(\epsilon_i, \Delta f_L(x_i)),$$

namely,

$$\frac{\Pr[\mathcal{T}_i(x_i) \in s_i]}{\Pr[\mathcal{T}_i(x_i') \in s_i]} \leq \exp(\epsilon_i \times \Delta f_L(x_i)).$$

Then according to [5], the parallel composition of $\mathcal{T}_i$s will have,

$$\Pr[\mathcal{T}(X) \in S] = \prod_{i \in \mathcal{U}} \Pr[\mathcal{T}_i(x_i) \in s_i]$$

Therefore,

$$\frac{\Pr[\mathcal{T}(X) \in S]}{\Pr[\mathcal{T}(X') \in S]} = \prod_{i \in \mathcal{U}} \frac{\Pr[\mathcal{T}_i(x_i) \in s_i]}{\Pr[\mathcal{T}_i(x_i') \in s_i]}$$

$$= \prod_{i \in \mathcal{U}} \exp\left(\epsilon_i \times \Delta f_L(x_i)\right) = \exp\left(\sum_{i \in \mathcal{U}} \epsilon_i \times \Delta f_L(x_i)\right)$$

Algorithm 1 Differentially Private Data Aggregation.

Input:

Original data curator's database: $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$;
Data aggregation function: f;
Local differential privacy budget, ϵ_i, for each data curator i.

Output:

Differentially private data aggregation output: $\mathcal{R}$.

1: Data curator i generates differentially private noise DP_i by ϵ_i and its local function sensitivity $\Delta f_L(x_i)$, i.e., $DP_i = DP(\epsilon_i, \Delta f_L(x_i))$, $\forall i \in \mathcal{U}$;
2: Data curator i sends $\mathcal{T}(x_i) = f(x_i) + DP(\epsilon_i, \Delta f_L(x_i))$ to the data aggregator, $\forall i \in \mathcal{U}$;
3: Data aggregator computes $\mathcal{R} = f(\mathcal{T}(x_1), \mathcal{T}(x_2), \ldots, \mathcal{T}(x_n))$;
4: **return** $\mathcal{R}$ to all data curators;

$$
\begin{aligned}
&= \exp\left(\sum_{i\in\mathcal{U}} \frac{\epsilon_i \times \Delta f_L(x_i)}{f(x_i)} \times f(x_i)\right) \\
&\leq \exp\left(\max_{i\in\mathcal{U}}\left\{\frac{\Delta f_L(x_i)}{f(x_i)}\epsilon_i\right\} \times \sum_{i\in\mathcal{U}} f(x_i)\right) \\
&\leq \exp\left(\max_{i\in\mathcal{U}}\left\{\frac{\Delta f_L(x_i)}{f(x_i)}\epsilon_i\right\} \times |\mathcal{U}| \times \max_{i\in\mathcal{U}}\{f(x_i)\}\right) \\
&= \exp\left(\max_{i\in\mathcal{U}}\left\{\frac{|\mathcal{U}|\cdot\Delta f_L(x_i)}{f(x_i)}\epsilon_i\right\} \times \Delta f_L(X)\right)
\end{aligned}
$$

Let the ϵ in Equation (4) be $\max_{i\in\mathcal{U}}\left\{\frac{|\mathcal{U}|\cdot\Delta f_L(x_i)}{f(x_i)}\epsilon_i\right\}$, then we prove the parallel composition of $\mathcal{T}_i$ is $\max_{i\in\mathcal{U}}\left\{\frac{|\mathcal{U}|\cdot\Delta f_L(x_i)}{f(x_i)}\epsilon_i\right\}$-differentially private, if $\mathcal{T}_i$ is ϵ_i-differentially private. ■

In fact, since we relaxed the condition that $\Delta f_L(X) \geq \sum_{i\in\mathcal{U}} \Delta f_L(x_i)$ in [5], we can treat our result as an approximation of the lower bound of privacy budget for distributed differential privacy in [5]. Furthermore, we use the local sensitivity $\Delta f_L(X)$ which is much smaller than the global sensitivity $\Delta f_G(X)$, therefore we provide better data utility than [6].

Finally, Algorithm 1 shows the way to guarantee the distributed differential privacy for summation (counting) in a star network.

V. Privacy Performance of Distributed Differential Privacy with Data Updates

In this section, we will study the privacy performance (i.e., lower bound of privacy budget ϵ for global differential privacy) of Algorithm 1 when we update data in some real-world applications. Specifically, our discussion will mainly focus on two types of data updates: passive data updates (e.g., malicious data cleansing) and active data updates (e.g., dynamically adding/removing data curator, serially refreshing data).

Case 1: Passive Data Updates. In the real-world applications, to mislead the analysis of data aggregation, a malicious attacker i may arbitrarily pollute his/her original data $f(x_i)$ as $f'(x_i) = f(x_i) + \delta$, where δ is the arbitrary pollution. In this part, we will discuss the lower bound of privacy budget for a distributed differentially private data summation (or counting) function (Figure 1) after cleansing the polluted data. Particularly, there are two sub-cases should be considered, (1) the malicious adversaries pollute their data and follow the protocol in Alg 1; (2) the malicious adversaries pollute their data but refuse to follow the differential privacy protocol in Alg 1. Next, we will study the two cases respectively, where we name the set of malicious data curators as $\mathcal{M}$, the set of non-malicious data curators as $\mathcal{H}$, the set of all data curators as $\mathcal{U} = \mathcal{M} \cup \mathcal{H}$, where $\mathcal{M} \cap \mathcal{H} = \varnothing$.

Sub-case 1. In this case, once we cleansed the polluted data, the information submitted by a malicious adversary i would also be $\mathcal{T}_i(x_i) = f(x_i) + DPnoise(\epsilon_i, \Delta f_L(x_i))$, $\forall i \in \mathcal{M}$, no matter the adversary adds differentially private noise before or after polluted data injection. Therefore, the lower bound of ϵ for global differential privacy will be guaranteed by Lemma 1.

Sub-case 2. After cleansing the polluted data (arbitrary value), the data submitted by a malicious data curator j, who did not provide any differential privacy guarantee on the submitted data, will be $\mathcal{T}_j(x_j) = f(x_j)$. In this case, we will have Lemma 2 to show the lower bound of privacy budget ϵ in such case.

Lemma 2: Let $\mathcal{T}_i$ provide ϵ_i-differential privacy, $\forall i \in \mathcal{H}$, $\mathcal{T}_j$ does not provide differential privacy, $\forall j \in \mathcal{M}$, then the parallel composition of all $\mathcal{T}_i$ and $\mathcal{T}_j$ will not have differential privacy guarantee.

Proof: $\because \forall j \in \mathcal{M}$, $\mathcal{T}_j$ does not provide differential privacy, $\therefore \forall j \in \mathcal{M}$, $\mathcal{T}_j(x_j) = f(x_j) = f(x_j) + 0 = f(x_j) + DPnoise(\epsilon_j, \Delta f_L(x_j))$. According to Definition 1, $\epsilon_j = +\infty > \epsilon_i$, $\forall i \in \mathcal{H}$. Then based on Lemma 1, the parallel composition of all $\mathcal{T}_i$ and $\mathcal{T}_j$ will be $\{+\infty\}$-differentially private. Namely, we do not have any differential privacy guarantee. ■

Therefore, in Case 1, to guarantee distributed differential privacy in a data aggregation (applying summation/counting function) system (in Figure 1), we must assume that the malicious adversaries only arbitrarily pollute the data but faithfully follow the local differential privacy protocol.

Case 2: Active Data Updates. In the real-world applications, we may have the following cases: new data curators join in the distributed system, existing data curators leave the system, and the data curators regularly refresh their private database. In Lemma 3, we will study the lower bound of privacy budget in such cases of active data updates.

Lemma 3: If the current parallel composition of $\mathcal{T}_i$ provides $\max_{i\in\mathcal{U}}\left\{\frac{|\mathcal{U}|\cdot\Delta f_L(x_i)}{f(x_i)}\epsilon_i\right\}$-differential privacy, then adding or removing a $\mathcal{T}_j$, refreshing data by a mechanism

$\mathcal{T}_i$ will not impact such differential privacy guarantee, unless there is a $\mathcal{T}_k$ does not provide differential privacy after an update.

Proof: Assume the existing differentially private mechanism $\mathcal{T}$ satisfies:

$$\frac{\Pr[\mathcal{T}(X) \in S]}{\Pr[\mathcal{T}(X') \in S]} \leq \exp\left(\max_{i \in \mathcal{U}} \left\{ \frac{|\mathcal{U}| \cdot \Delta f_L(x_i)}{f(x_i)} \epsilon_i \right\} \times \Delta f_L(X)\right)$$

where $\mathcal{U}$ is the set of all existing data curators.

Then when we add a $\mathcal{T}_j$ who provides ϵ_j-differential privacy, the parallel composition of the current $\mathcal{T}$ and $\mathcal{T}_j$ will be,

$$\begin{aligned}
&\frac{\Pr[\mathcal{T}(X) \in S]}{\Pr[\mathcal{T}(X') \in S]} \times \frac{\Pr[\mathcal{T}_j(x_j) \in s_j]}{\Pr[\mathcal{T}_j(x'_j) \in s_j]} \\
\leq &\exp\left(\max_{i \in \mathcal{U}} \left\{ \frac{|\mathcal{U}| \cdot \Delta f_L(x_i)}{f(x_i)} \epsilon_i \right\} \times \Delta f_L(X)\right) \\
&\times \exp(\epsilon_j \times \Delta f_L(x_j)) \\
= &\exp\left(\max_{i \in \mathcal{U}} \left\{ \frac{\Delta f_L(x_i)}{f(x_i)} \epsilon_i \right\} \times |\mathcal{U}| \times \max_{i \in \mathcal{U}} f(x_i)\right) \\
&\times \exp\left(\frac{\Delta f_L(x_j)}{f(x_j)} \epsilon_j \times f(x_j)\right) \\
\leq &\exp\left(\max_{i \in \{\mathcal{U}+j\}} \left\{ \frac{\Delta f_L(x_i)}{f(x_i)} \epsilon_i \right\} \times (|\mathcal{U}|+1) \times \max_{i \in \{\mathcal{U}+j\}} f(x_i)\right) \\
= &\exp\left(\max_{i \in \mathcal{U}'} \left\{ \frac{|\mathcal{U}'| \cdot \Delta f_L(x_i)}{f(x_i)} \epsilon_i \right\} \times \Delta f_L(X)\right)
\end{aligned}$$

Since the new data curator set $\mathcal{U}'$ contains all the data curators including the new joined one, the conclusion from Lemma 1 will be held. Similarly, we can prove the case of removing a $\mathcal{T}_j$ by simply replacing $\times$ to $\div$, and $+$ to $-$. Moreover, we can treat data refresh as a sequential composition of addition and deletion, then prove its lower bound of privacy budget by the above in two steps. ■

Based on Lemma 1, Lemma 2, and Lemma 3, we will have Theorem 1.

Theorem 1: Let a mechanism $\mathcal{T}_i$, $i \in \mathcal{U}$, provide ϵ_i-differential privacy, then a parallel composition of $\mathcal{T}_i$ will guarantee $\max_{i \in \mathcal{U}} \left\{ \frac{|\mathcal{U}| \cdot \Delta f_L(x_i)}{f(x_i)} \epsilon_i \right\}$-differential privacy in case of distributed summation (counting), cleansing polluted data, dynamically adding or removing data curators, and refreshing data, unless there is a malicious mechanism $\mathcal{T}_j$ refuses to guarantee differential privacy on its original database.

VI. Experimental Evaluation

In this section, we will use two real-world datasets to evaluate the performance of the lower bound of privacy budget ϵ for distributed differential privacy. We start by introducing the two real-world datasets and evaluation metric, then show the results of the experiments over [5], [6] and our work on the real-world datasets.

A. Datasets

We use two real-world datasets for the experimental evaluation in this paper: MovieLens 100K dataset[2] and Electricity Load Diagrams 2011-2014 dataset[3]. Specifically, in the MovieLens 100K dataset, 943 users posted 100,000 ratings on 1682 films. While, in the Electricity Load Diagrams 2011–2014 dataset, there are 370 electricity meters reported the energy consumption every 15 minutes for four years from 2011 to 2014.

Table II is an example of MovieLens 100K dataset, where the values are the ratings from user x_i on film y_j. That is, there is no such user who always gives high rate on movies. Table III is an example of Electricity Load Diagrams 2011–2014 dataset, where the values are the electricity consumption read by meter x_i between time y_{j-1} and y_j. Namely, the one which reads larger number will always do so.

Table II: An Example of ML-100 dataset.

	x_1	x_2	x_3
y_1	1	4	1
y_2	1	1	4
y_3	5	2	1

Table III: An Example of ELD-1114 dataset.

	x_1	x_2	x_3
y_1	51	10	100
y_2	62	11	110
y_3	61	22	105

B. Evaluation Metric, and Experiment Settings

In the experiments, we concurrently inject differentially private noises into the distributed system at the end of data aggregator (by $\Delta f_L(X)$ and ϵ) and data curators (by $\Delta f_L(x_i)$ and ϵ_i). Then a smaller difference between the overall noises from the two ends indicates the corresponding lower bound of the global differential privacy (ϵ) is better. To measure the difference, we use Mean Absolute Error (MAE):

$$MAE = \frac{1}{N} \left| \sum_{i=1}^{N} (|G_i| - |L_i|) \right|,$$

where N is the total rounds of experiments, L_i is the overall noise added by data curators at experiment round i (local differential privacy), G_i is the overall noise added by the lower bound of global privacy budget (global differential privacy). We set N = 100K. Clearly, a better lower bound of global privacy budget will have a smaller value of MAE. Table IV shows the parameters we used to evaluate different

[2]https://grouplens.org/datasets/movielens/100k/

[3]https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

Table IV: The parameters in the experiments.

	Curators	Aggregator
SumEpsilon [6]	$\Delta f_G(X)$, ϵ_i	$\Delta f_G(X)$, $\sum_i \epsilon_i$
MaxEpsilon [5]	$\Delta f_L(x_i)$, ϵ_i	$\Delta f_L(X)$, $\max_i\{\epsilon_i\}$
ThisWork	$\Delta f_L(x_i)$, ϵ_i	$\Delta f_L(X)$, $\max_i \left\{ \frac{\lvert\mathcal{U}\rvert \cdot \Delta f_L(x_i)}{f(x_i)} \epsilon_i \right\}$

lower bounds of global privacy budget of differential privacy.

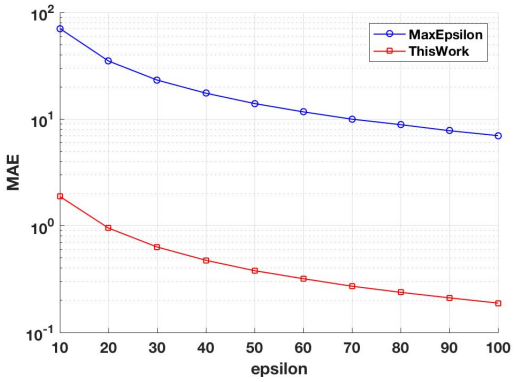
C. Experimental Results Analysis

In the figures for the experimental results, we rename the MovieLens 100K dataset as ML-100, the Electricity Load Diagrams 2011-2014 dataset as ELD-1114.

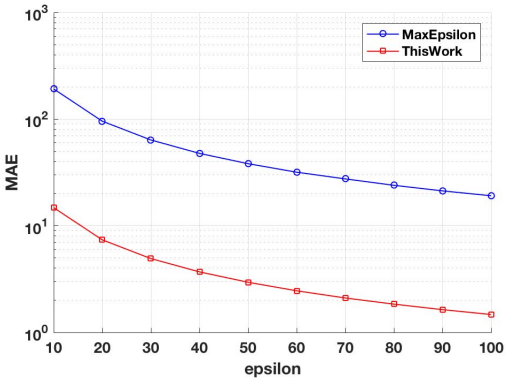
Figure 2 shows the privacy performance of different lower bounds of global privacy budget on the MovieLens 100K dataset, where f is counting in Figure 2a and summation in Figure 2b. As we know, in ML-100 dataset, $\Delta f_L(X) \leq \sum_{i\in\mathcal{U}} f_L(x_i)$ (e.g., when f is counting, $\Delta f_L(X) = 730$, $\sum_{i\in\mathcal{U}} f_L(x_i) = 943$). So in this case, the lower bound in [5] will not be guaranteed. Moreover, from both Figure 2a and Figure 2b, the noises injected by our lower bound of global ϵ are closer to the real noises injected into the system than MaxEpsilon. So our privacy budget provides better privacy performance than [5] in the case that $\Delta f_L(X) \leq \sum_{i\in\mathcal{U}} f_L(x_i)$.

Figure 3 shows the privacy performance (on summation function) by different lower bounds of global privacy budget on Electricity Load Diagrams dataset. In the ELD-1114 dataset, $\Delta f_L(X) \geq \sum_{i\in\mathcal{U}} f_L(x_i)$, so according to Theorem 4 in [5] and Theorem 1 in this paper, the privacy performance of MaxEpsilon would be better than ours. In fact, Figure 3 clearly shows the above statement. Furthermore, if we compare the experimental results in Figure 2 and Figure 3, we can find that in the cases where MaxEpsilon [5] does not work, the privacy performance of our global privacy budget is much better than [5]. While in the cases where MaxEpsilon [5] provides $\max_{i\in\mathcal{U}} \epsilon_i$-distributed differential privacy correctly, the privacy performance of our privacy budget is close to MaxEpsilon [5]. Therefore, as an approximation of MaxEpsilon [5], the overall privacy performance of our privacy budget is satisfied. But since our privacy budget does not rely on any condition, the lower bound of privacy budget in this paper is more adaptable in the general cases.

In Figure 4, we compare the noises injected from the local differential privacy by SumEpsilon and ThisWork (MaxEpsilon) on the MovieLens 100K dataset and the Electricity Load Diagrams dataset, where f is summation function. Same as our analysis in Section IV, both Figure 4a and Figure 4b shows that the noises injected by the schemes in [6] (using $\Delta f_G(X)$ to generate differentially private noise) is much greater than MaxEpsilon [5] and ours (using $\Delta f_L(X)$ to generate differentially private noise).



(a) Counting



(b) Summation

Figure 2: Privacy Performance on ML-100.

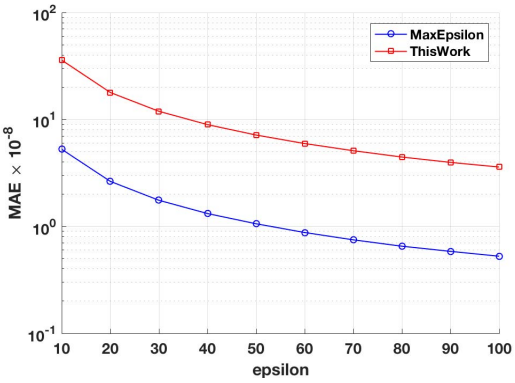
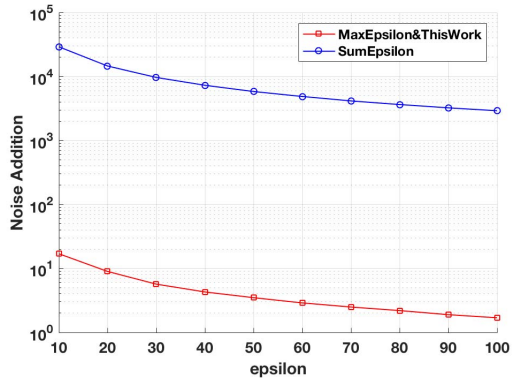


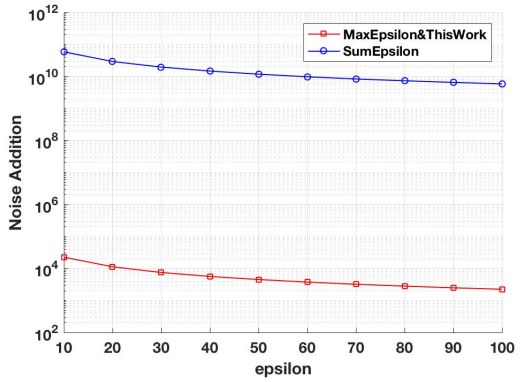
Figure 3: Privacy Performance on ELD-1114 (summation).

VII. Conclusion

In the recent years, we take the advantages of sensor networks to collect and analyse data for better decision making. Among all the data analysis functions, summation (counting) function is the most important one. To preserve individual privacy against the worst collusion attacks, the existing schemes either provides low data utility by using a



(a) ML-100



(b) ELD-1114

Figure 4: Noise Injection by Different Function Sensitivity for Local Differential Privacy.

global sensitivity of summation (counting) function or unknown privacy guarantee in some special cases. To address such problems, in this paper, we study the privacy property of distributed differential privacy in different scenarios to offer a new lower bound of differential privacy budget with unconditional function sensitivity in a distributed system. Both theoretical and experimental analysis show that our lower bound of differential privacy budget for a distributed system guarantees better privacy performance than the existing solutions.

ACKNOWLEDGMENT

This work is supported by Australian Government Research Training Program Scholarship, Australian Research Council Discovery Project DP150104871, and Research Initiative Grant of Sun Yat-sen University under Project 985. The corresponding author is Hong Shen.

REFERENCES

[1] A. C. Yao, "Protocols for secure computations," in *Foundations of Computer Science, 1982. SFCS'82. 23rd Annual Symposium on*. IEEE, 1982, pp. 160–164.

[2] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 05, pp. 557–570, 2002.

[3] C. Dwork, "Differential privacy," in *Automata, languages and programming*. Springer, 2006, pp. 1–12.

[4] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2006, pp. 486–503.

[5] F. McSherry, "Privacy integrated queries," in *Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data*. ACM, 2009.

[6] E. Shi, H. Chan, E. Rieffel, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *Annual Network & Distributed System Security Symposium (NDSS)*. Internet Society., 2011.

[7] M. Pettai and P. Laud, "Combining differential privacy and secure multiparty computation," in *Proceedings of the 31st Annual Computer Security Applications Conference*. ACM, 2015, pp. 421–430.

[8] J. Cao, F.-Y. Rao, E. Bertino, and M. Kantarcioglu, "A hybrid private record linkage scheme: separating differentially private synopses from matching records," in *2015 IEEE 31st International Conference on Data Engineering*. IEEE, 2015, pp. 1011–1022.

[9] Y. Hong, J. Vaidya, H. Lu, P. Karras, and S. Goel, "Collaborative search log sanitization: Toward differential privacy and boosted utility," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 5, pp. 504–518, 2015.

[10] E. Nozari, P. Tallapragada, and J. Cortés, "Differentially private distributed convex optimization via objective perturbation," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 2061–2066.

[11] Y. Xiao, L. Xiong, and C. Yuan, "Differentially private data release through multidimensional partitioning," in *Workshop on Secure Data Management*. Springer, 2010, pp. 150–168.

[12] N. Mohammed, R. Chen, B. Fung, and P. S. Yu, "Differentially private data release for data mining," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 493–501.

[13] Z. Lu and H. Shen, "A security-assured accuracy-maximised privacy preserving collaborative filtering recommendation algorithm," in *Proceedings of the 19th International Database Engineering & Applications Symposium*. ACM, 2015, pp. 72–80.

[14] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Theory of cryptography*. Springer, 2006, pp. 265–284.

[15] K. Nissim, S. Raskhodnikova, and A. Smith, "Smooth sensitivity and sampling in private data analysis," in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM, 2007, pp. 75–84.