Computing Aggregates over Numeric Data with Personalized Local Differential Privacy

Mousumi Akter[1] (✉) and Tanzima Hashem[2]

Department of Computer Science and Engineering,
Bangladesh University of Engineering and Technology, Dhaka, Bangladesh
[1]mousumiakter22@gmail.com
[2]tanzimahashem@cse.buet.ac.bd

Abstract. The advancement of technology and the widespread usage of smart phones have made the collection of data from users easy and cost-effective, which allows the government, urban planner, and researchers to envision novel analysis. Along with the benefits, the shared data can bring serious privacy concerns as they reveal sensitive information about a user. Differential privacy has become an effective model for sharing privacy protected data with others. To facilitate users to protect the privacy of data before it leaves their personal devices, the concept of personal local differential privacy (PLDP) has been introduced for counting queries. We formulate PLDP for computing aggregates over numeric data. We present an efficient approach, private estimation of numeric aggregates (PENA), that guarantees PLDP of numeric data while computing an aggregate (e.g., the average or the minimum). We perform extensive experiments over a real dataset to show the effectiveness of PENA.

1 Introduction

In this era of flourishing Internet and smart phones, data collection from users has become easier and cost-effective and opened the door for novel applications and analysis. Business models like Waze - GPS, Maps and Traffic [1] have already been established based on user data. Not only in business, data collection has a huge impact on research; agglomeration of data and its analysis help researchers to perceive answers of their research questions and hypotheses. On the other hand, to grasp the behavior of a community there is no substitute for the data collection. Thus, the collection of enormous amount of real-time and historical data from users allows the government, business, and researchers to contribute in different domains for the improvement of the quality of human lives.

Along with the benefits, sharing data with others may bring serious privacy concerns as a user's data can reveal sensitive and private information about the user's health, habit and preference. Considering the privacy issues, traditionally data is shared with trusted parties, who are responsible for ensuring the privacy of user data before sharing the data with others. However, unexpected leakage of

[1] https://www.waze.com

personal data from the trusted authority may also cause a massive devastation, which happened to *Netflix*[2] and *AOL*[3]. In this paper, we aim to develop a novel approach to ensure the personalized local differential privacy (PLDP) of numeric data before it leaves a user's device. We focus on computing aggregate statistics over private numeric data collected from users in a distributed manner.

Differential privacy [7] is a widely accepted framework developed for ensuring data privacy of a statistical database. Protecting differential privacy of time series and numeric data in the centralized setting has been studied in the literature [17, 18]. In a centralized setting, users provide data to a central trusted authority, and the trusted authority protects user data from others by applying the concept of differential privacy. Differential privacy adds noise to the data to provide rigorous privacy guarantee with an accuracy bound. However, in the local setting, users do not even trust the central authority [3] and want their data to be protected before leaving their devices without knowing the data of other users. Therefore, the local differential privacy is rigid to achieve and also challenging. In case of the centralized differential privacy, data from all users need to be aggregated first and then the noise is added to the data using Laplace or exponential mechanism according to the sensitivity [5] so that a user's data is not identifiable with a certain confidence level in the computed aggregate statistics. In the local setting, data from all users are not aggregated at a single place and thus, the traditional definition of the differential privacy is not applicable and the concepts of using Laplace and exponential mechanisms do not apply as they failed to achieve the desired level of accuracy [3].

Differential privacy in the local setting is termed as Local Differential Privacy (LDP) [15]. LDP paves a better way to achieve privacy beyond trusting the central authority and other users. Few recent works [9, 15] have been done to ensure LDP of numeric data. For LDP, a user shares a randomized value instead of the actual one for a numeric attribute with an accumulator such that no one can reverse engineer the actual value from the shared data with a certain confidence level. Specifically, the confidence level is expressed as the maximum ratio of the probabilities of computing the shared randomized values for any pair of values of the numeric attribute. A major limitation of LDP is that if the possible range of the values of a numeric attribute is large, the accuracy of the computed aggregates over shared randomized values degrades significantly. However, in reality, people may have background knowledge about the range of possible values for an attribute. For example, though a salary attribute can have any positive numeric value but in reality, people may know the range of salaries depending on the workplace where a user is employed. Furthermore, users may not need to have the same privacy in terms of the confidence level but LDP does not provide flexibility to users to set their privacy levels, i.e., LDP assumes all users have same privacy level [15].

To overcome the above limitations of LDP, recently, personalized local differential privacy (PLDP) [3] has been introduced that gives users the flexibility

[2] http://money.cnn.com/galleries/2010/technology/1012/gallery.5_data_breaches
[3] https://en.wikipedia.org/wiki/AOL_search_data_leak

to control their privacy levels. However, their work is limited to the counting query and can not be extended for an aggregate function as they use one bit protocol [2]. In one bit protocol, a random bit is sent using Bernoulli probability distribution from the user and by tracing this bit answers of different histogram queries are estimated (e.g., how many people in a community likes to go for shopping on Sunday?).

In our research problem, we compute statistical aggregates such as finding the summation or the minimum values over numeric data, where identities of users are revealed for the authorization purpose but the privacy of the data shared by the users is protected. The accumulator knows who are taking part in the system but does not know what is the actual data of a user. Hence, obscuring user data from the central authority and from other users while facilitating the computation of aggregate functions is our main challenge. To the best of our knowledge, there is no work that ensures PLDP of numeric data and can compute any aggregate function.

In this paper, we propose a novel approach, private estimation of numeric aggregates (PENA) to compute aggregates over numeric data while ensuring PLDP. The underlying idea of PENA is to collect random responses from users over a safe range, i.e., the range within which a user's data is not identified with a specified confidence level. We develop a Local Random Responser (LRR) that generates a random response while ensuring PLDP of a user's data using Bernoulli probability distribution [3]. Bernoulli probability distribution ensures both the utility of responses and privacy of users.

In summary, the contributions of the paper are as follows:

- We formulate PLDP for computing aggregate functions over numeric data.
- We present an efficient approach PENA that can guarantee PLDP of numeric data while computing an aggregate function.
- We present the theoretical proof of the correctness of our solution.
- We perform extensive experiments to show that the effectiveness of our proposed approach.

The remainder of this paper is organized as follows. Section 2 formulates the problem, discusses the threat model, and shows the system architecture. In Section 3, we present our approach, PENA. Section 4 presents the results of our evaluation of PENA using real datasets. In Section 5, we discuss the related work. Finally, Section 6 concludes the paper.

2 Problem Formulation

We first formally define the concepts of differential privacy (DP), local differential privacy (LDP), and personalized local differential privacy (PLDP), and then formulate the problem that we consider in this paper. We discuss our threat model in Section 2.1 and the system architecture in Section 2.2.

Differential Privacy (DP) [5, 8]. A randomized function f provides ϵ-differential privacy if for two databases t, t' that differs from at most one row and for all $z \in Z$, $\frac{Pr[f(t) \in z]}{Pr[f(t') \in z]} \leq e^{\epsilon}$.

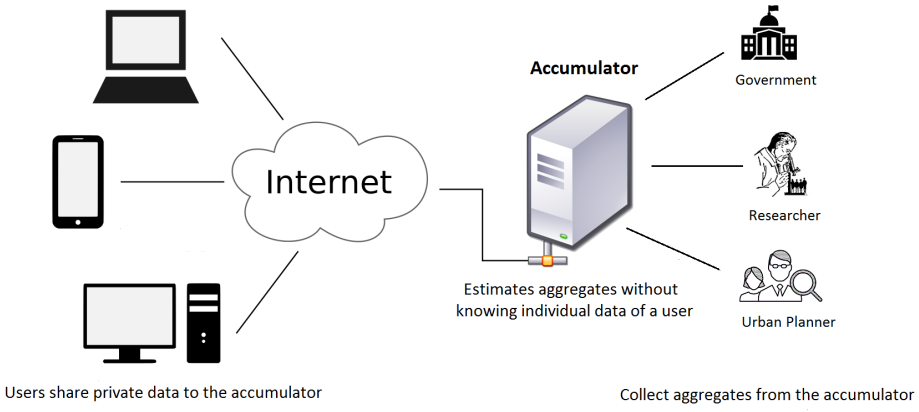


Fig. 1. System Architecture.

Local Differential Privacy (LDP) [15]. A randomized function f provides ϵ-local differential privacy, if and only if for any two values of an attribute t, $t' \in$ Dom (f) and for any possible output t^* of f, $\frac{Pr[f(t)=t^*]}{Pr[f(t')=t^*]} \leq e^{\epsilon}$.

Personalized Local Differential Privacy (PLDP) [3]. A randomized function f provides $(\mathcal{T}, \epsilon)$-personalized local differential privacy, if and only if for any two values of an attribute t, $t' \in \tau$ and for any possible output t^* of f, $\frac{Pr[f(t)=t^*]}{Pr[f(t')=t^*]} \leq e^{\epsilon}$.

In the case of PLDP, $\mathcal{T}$ defines a safe range for a user and each user can have her own privacy requirement in term of ϵ. For example, if the numeric data of any user u is \$700, the user may feel safe to share the data in the range $\mathcal{T}$ = \$0 - \$10000. If ϵ=0.2, it means that the ratio of the probabilities of generating t^* is less than or equal to $e^{0.2}$.

In this paper, we address the problem of ensuring PLDP for numeric data of users while computing aggregate functions like the average or the minimum. Formally, given a group of n users $U = \{u_1, , u_2, \ldots, u_n\}$, a numeric data $t \in \mathcal{T}$ of every user in the group transformed according to the privacy specification $(\mathcal{T}, \epsilon)$ of the user, the accumulator computes the aggregates over the shared private data of users.

2.1 Threat Model

We consider the accumulator, other users, and eavesdroppers as adversaries. Users do not want their numeric data to be identified in a safe range with more than the required confidence level. The target of adversaries is to identify the actual numeric data of users, refine the safe range and increase the confidence level of identifying the numeric data. We assume that users and the accumulator follow the protocol of the system while sharing their numeric data and computing aggregates.

2.2 System Architecture

Users are connected to an accumulator through the Internet or wireless adhoc networks. Figure 1 shows the system architecture of our proposed approach. Every user independently shares their data after ensuring their PLDP. The accumulator then generates the aggregates (e.g., the average or the minimum) and provides them to the government, researchers, urban planners, and others.

3 Private Estimation of Numeric Aggregates (PENA)

In this Section, we present our approach, private estimation of numeric aggregates (PENA) to compute aggregates over numeric data while ensuring PLDP. PENA uses a local random responser (LRR) and exploits Bernoulli probability distribution [15] to achieve PLDP. Bernoulli probability distribution is initially designed to guarantee LDP in [15], where every user has same privacy level, i.e., same ϵ_i. In this paper, we extend it for ensuring PLDP, where users can have different ϵ_i.

Algorithm 1 shows the pseudocode for PENA. The general idea of our PENA framework is to collect random responses from users over a specified safe range $\mathcal{T}$. For different subsets of users $\mathcal{T}$ can be different. For simplicity, we assume here for a subset of users $\{u_1, u_2, \ldots, u_n\}$ have safe range $\mathcal{T}$. Every user u_i among n users responds with a random numeric value f_i that is generated using Bernoulli probability distribution using function LRR (Line 2). After receiving $f_1, f_2, \ldots, f_n$, the accumulator estimates the aggregate over the received values using Function $ComputeAggregate$ (Line 4). For example, if the aggregate is average then $ComputeAggregate$ estimates the aggregate as $(\mathcal{T}_{max} - \mathcal{T}_{min}) \times \frac{\sum_{i=1}^{n} f_i}{n}$, where $\mathcal{T}_{max}$ and $\mathcal{T}_{min}$ represent the maximum and minimum values of the safe range $\mathcal{T}$.

Algorithm 1 Private Estimation of Numeric Aggregates

Input: A group of n users $U = \{u_1, u_2, \ldots, u_n\}$
Output: Estimate numeric aggregates f over the random responses of users
1: **for** each user u_i **do**
2: $\quad f_i \leftarrow LRR(\mathcal{T}, t_i, \epsilon_i)$
3: **end for**
4: **return** $f \leftarrow ComputeAggregate(f_1, f_2, \ldots, f_n)$

In the next Section, we develop a local random responser (LRR) to achieve PLDP over numeric data.

3.1 Local Random Responser (LRR)

We use Bernoulli probability distribution to develop an LRR, which has been already shown in the literature [15] as an effective way to ensure LDP of numeric

data. Algorithm 2 shows the pseudocode for function *LRR*. Each user u_i runs *LRR* to compute a randomized value t_i^* based on her actual value t_i, safe range $\mathcal{T}$, and privacy parameter ϵ_i. The output of the algorithm is t_i^*.

A user first scales t_i to [-1, 1] using safe range $\mathcal{T}$ (Line 1 of Algorithm 2). For example, if a user's data is \$5700 and the safe range is \$0-\$10000, the user scales the data to 0.14. Then the user randomizes t_i into a new response t_i^* using LRR and sends it to the accumulator. The value of t_i^* is generated using Bernoulli random variable in a way that satisfies ($\mathcal{T}$, ϵ_i)-PLDP for user u_i (proof is presented in Section 3.2). LRR generates two types of random responses. The probability of generating a random response among two options is calculated using a Bernoulli random variable, and the calculation of the probability depends on the user's scaled value t_i and defined confidence level ϵ_i (Line 2). It can be compared with a coin flip. The probability of generating head is calculated by Bernoulli random variable. Then the coin is flipped with the computed probability. If head is found then the user responds with $\frac{e^{\epsilon_\mathrm{i}}+1}{e^{\epsilon_\mathrm{i}}-1}$ (Line 4). Otherwise, the user responds with $-\frac{e^{\epsilon_\mathrm{i}}+1}{e^{\epsilon_\mathrm{i}}-1}$ (Line 6).

Algorithm 2 Local Random Responser

Input: Safe range $\mathcal{T}$
Input: User u_i's numeric data t_i
Input: User u_i's privacy parameter ϵ_i
Output: Randomized response $t_i^* \in \{\frac{e^{\epsilon_\mathrm{i}}+1}{e^{\epsilon_\mathrm{i}}-1}, -\frac{e^{\epsilon_\mathrm{i}}+1}{e^{\epsilon_\mathrm{i}}-1}\}$
1: Generate scaled $t_i \in$ [-1, 1] using $\mathcal{T}$
2: Sample a Bernoulli variable b such that $\Pr[b = 1] = \frac{t_i.(e^{\epsilon_\mathrm{i}}-1)+e^{\epsilon_\mathrm{i}}+1}{2e^{\epsilon_\mathrm{i}}+2}$
3: **if** $b = 1$ **then**
4: $\quad t_i^* \leftarrow \frac{e^{\epsilon_\mathrm{i}}+1}{e^{\epsilon_\mathrm{i}}-1}$
5: **else**
6: $\quad t_i^* \leftarrow -\frac{e^{\epsilon_\mathrm{i}}+1}{e^{\epsilon_\mathrm{i}}-1}$
7: **end if**
8: **return** t_i^*

3.2 Theoretical Analysis

In this Section, we present the theoretical analysis of privacy assurance of our proposed approach. We give the following theorem to prove that LRR guarantees PLDP for every user.

Theorem 1. *For any user u_i with privacy specification (τ, ϵ_i) and any $t_i^* \in \{\frac{e^{\epsilon}i+1}{e^{\epsilon}i-1}, -\frac{e^{\epsilon}i+1}{e^{\epsilon}i-1}\}$ LRR guarantees (τ, ϵ_i)-PLDP for u_i.*

Proof. By definition of PLDP, we have to prove that, for any t_i, $t_i' \in \tau$ and any $t_i^* \in \{\frac{e^{\epsilon_\mathrm{i}}+1}{e^{\epsilon_\mathrm{i}}-1}, -\frac{e^{\epsilon_\mathrm{i}}+1}{e^{\epsilon_\mathrm{i}}-1}\}$, $\frac{Pr[LRR(\mathcal{T},t_i,\epsilon_i)=t_i^*]}{Pr[LRR(\mathcal{T},t_i',\epsilon_i)=t_i^*]} \leq e^{\epsilon_\mathrm{i}}$.

LRR scales t_i to [-1, 1] using safe range $\mathcal{T}$, and assume that t_i^* is $\frac{e^{\epsilon_i}+1}{e^{\epsilon_i}-1}$. For, the other case, i.e., $t_i^* = -\frac{e^{\epsilon_i}+1}{e^{\epsilon_i}-1}$, the proof can be done similarly.

According to Algorithm 2, the probabilities to compute $t_i^* = \frac{e^{\epsilon_i}+1}{e^{\epsilon_i}-1}$ for t_i and t_i' are $\Pr[\text{LRR}(\mathcal{T}, t_i, \epsilon_i)=t_i^*]=\frac{t_i.(e^{\epsilon_i}-1)+e^{\epsilon_i}+1}{2e^{\epsilon_i}+2}$ and $\Pr[\text{LRR}(\mathcal{T}, t_i', \epsilon_i)=t_i^*]=\frac{t_i'.(e^{\epsilon_i}-1)+e^{\epsilon_i}+1}{2e^{\epsilon_i}+2}$, respectively.

Thus,

$$\begin{aligned}
\frac{Pr[LRR(\mathcal{T},t_i,\epsilon_i)=t_i^*]}{Pr[LRR(\mathcal{T},t_i',\epsilon_i)=t_i^*]} &= \frac{\frac{t_i.(e^{\epsilon_i}-1)+e^{\epsilon_i}+1}{2e^{\epsilon_i}+2}}{\frac{t_i'.(e^{\epsilon_i}-1)+e^{\epsilon_i}+1}{2e^{\epsilon_i}+2}} \\
&= \frac{t_i.(e^{\epsilon_i}-1)+e^{\epsilon_i}+1}{t_i'.(e^{\epsilon_i}-1)+e^{\epsilon_i}+1} \\
&\leq \frac{max_{t_i\in[-1,1]}(t_i.(e^{\epsilon_i}-1)+e^{\epsilon_i}+1)}{min_{t_i'\in[-1,1]}(t_i'.(e^{\epsilon_i}-1)+e^{\epsilon_i}+1)} \\
&\leq \frac{1.(e^{\epsilon_i}-1)+e^{\epsilon_i}+1}{-1.(e^{\epsilon_i}-1)+e^{\epsilon_i}+1} \\
&\leq \frac{e^{\epsilon_i}-1+e^{\epsilon_i}+1}{-e^{\epsilon_i}+1+e^{\epsilon_i}+1} \\
&\leq \frac{2e^{\epsilon_i}}{2} \\
&\leq e^{\epsilon_i}
\end{aligned}$$

We have $\frac{Pr[LRR(\mathcal{T},t_i,\epsilon_i)]}{Pr[LRR(\mathcal{T},t_i',\epsilon_i)]} \leq e^{\epsilon_i}$. Hence, LRR guarantees (τ, ϵ_i)-PLDP for u_i.

3.3 Simulation

In this Section, we illustrate our proposed approach PENA with an example. Suppose the accumulator sets the safe range $\mathcal{T}$ as \$0-\$10000. Users scale their data to [-1, 1] using the safe range $\mathcal{T}$, and generate random responses using local random responser (LRR). Without loss of generality, we show how a user computes her random response. Let the actual numeric data of a user is \$800 and ϵ is 0.2. The scaled numeric data of the user is $t=(\frac{800}{10000})*2\text{-}1 = \text{-}0.84$.

Since, $\frac{e^{\epsilon}+1}{e^{\epsilon}-1} = 10.03$ and $\frac{t.(e^{\epsilon}-1)+e^{\epsilon}+1}{2e^{\epsilon}+2} = 0.46$, the user either sends 10.03 with probability 0.46 or sends -10.03 with probability 0.54 (1-0.46) to the accumulator. Similarly, other users send their random responses to the accumulator.

The accumulator estimates the aggregate (e.g., the average or the minimum) over the received values from users and the safe range.

4 Experiments

In this Section, we evaluate and compare the performance of our proposed approach PENA through extensive experiments. Since there is no existing work for PLDP over numeric data in the literature, we modify the work [15] that is proposed for ensuring LDP of numeric data while computing aggregates and compare it with PENA. For LDP, the safe range $\mathcal{T}$ does not exist and $\mathcal{T}$ is assumed to be the set of all possible values for numeric data and thus, the achieved level of the accuracy for the computed aggregates is not satisfactory to apply in

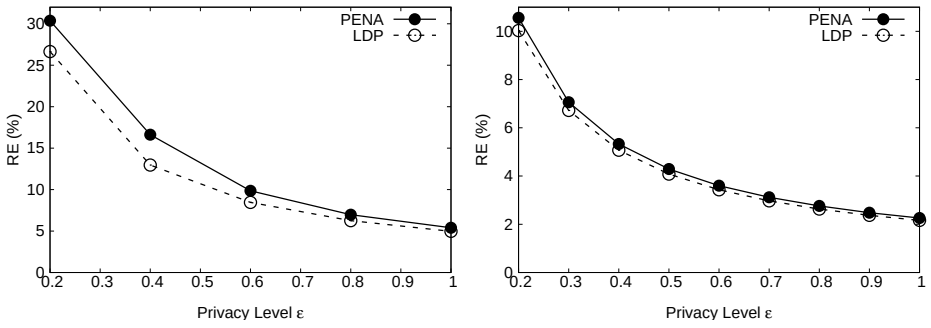
Table 1. Parameter Settings for Experiments

Parameter	Values	Default
Privacy level ϵ	0.2, 0.4, 0.6, 0.8, 1	0.5
User participation (%)	20, 40, 60, 80, 100	50
Safe range $\mathcal{T}$	1.0, 1.1, 1.2, 1.3, 1.4, 1.5	1.0

real scenarios. For our experiments, we incorporate $\mathcal{T}$ in [15]. However, we cannot extend [15] to support personalize privacy level ϵ. Note that we select [15] for our comparison because it has been shown in the literature that [15] outperforms other LDP based approaches for numeric data like [9].

We show our experiments for aggregate functions average and minimum. Our approach is also applicable for other types of aggregates (e.g., maximum). We validate our proposed solution using the dataset:IPUMS [1] that contains 3.15M total family income records of United States. The whole data space is normalized to [-1, 1] using safe range $\mathcal{T}$. We performed several sets of experiments by varying the following parameters: privacy level ϵ, the percentage of user participation over 3.1M tuples, safe range $\mathcal{T}$.

Table 1 shows ranges and default values used for each parameter. To observe the effect of one parameter in an experiment others are kept in default values. For [15], ϵ=0.5 means we set ϵ to 0.5 for all users, and in PENA, ϵ=0.5 means users have the flexibility to generate any random privacy level from 0 to 0.5. All experiments are run on an Intel-CORE i3 Windows 7 machine. For each experiment, we perform 100 independent runs and take the average performance of this 100 independent runs. Experimental results show that PENA outperforms modified LDP based approach [15] in terms of accuracy while ensuring higher privacy levels for users for both aggregates average and minimum.

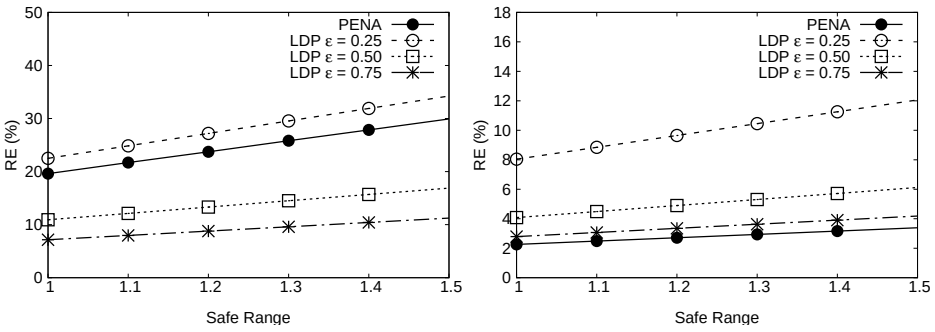


(a) average (b) minimum

Fig. 2. Effect of privacy Level (ϵ) on relative error (RE%)

Effect of Privacy Level (ϵ). Privacy level (ϵ) controls the privacy of a particular user. Figures 2(a) and 2(b) show that the relative error decreases with the relaxation of privacy level for both LDP and PENA. This is because,

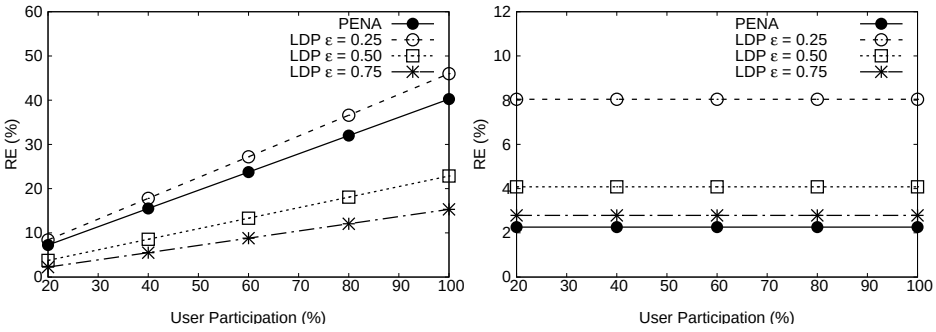
with the increase of ϵ, more accurate user data are used in the aggregation. On the other hand, users can have higher privacy levels in PENA than the LDP-based approach. For example, for ϵ=0.5, in PENA, values of ϵ for users are varied from 0 to 0.5, whereas in the LDP-based approach, all users have ϵ=0.5. Since a smaller value of ϵ ensures higher privacy for a user, most of the users in PENA have higher privacy levels than those in the LDP-based approach. In spite of ensuring the higher privacy level for users, both PENA and the LDP-based approach show similar levels of accuracy as shown in Figures 2.



(a) average (b) minimum

Fig. 3. Effect of safe range ($\mathcal{T}$) on relative error (RE%)

Effect of Safe Range ($\mathcal{T}$). A higher value of the safe range $\mathcal{T}$ ensures a higher level of privacy. We scale the maximum numeric value of the dataset to 1. Figure 3(a) and 3(b) show that relative error increases slowly for every (10%) increase of the safe range for average aggregation. For privacy level ϵ=0.30, PENA outperforms the LDP-based approach for aggregate function average (Figure 3(a)) and for ϵ=0.80, PENA outperforms the LDP-based approach for aggregate function minimum (Figure 3(b)).



(a) average (b) minimum

Fig. 4. Effect of user participation (%) on relative error (RE%)

Effect of Percentage of User Participation. Figure 4(a) shows that the relative error increases slightly with the increase of the percentage of user participation for aggregate function average. For 40% or less user participation (i.e., 1.26 M users among 3.15M), PENA generates error less than 20%. Figure 4(b) shows that the relative error remains almost constant over large dataset to evaluate minimum aggregate function. This result shows the effectiveness of PENA to handle large dataset to compute both aggregate functions average and minimum.

5 Related Work

Data privacy has been addressed in the literature using techniques like k-anonymity, perturbation, sampling, cryptography, secure multi-party computations and differential privacy. In the k-anonymity technique, a user's data is indistinguishable from the data of at least $k-1$ other users [11, 20]. Thus, a major limitation of the k-anonymity technique is that at least k users need to have the same data. In the perturbation technique [6, 12, 19], noise is added to the data without any theoretical guarantee of privacy. Sampling [6] based technique to ensure privacy only works well if the dataset is large and similarity exists in the data. Though cryptographic techniques [16] ensure strong privacy, they are not feasible for real world applications because of their extremely high processing overhead. Secure multi-party protocols [13] involve a group of users to compute aggregates, where a user's data privacy is violated if all group members collude. In recent years, differential privacy (DP) [7] has become an effective model to protect data privacy of users because of its theoretical privacy guarantee and less processing overhead.

DP has been introduced in [4] and since then it has been applied to solve variant problems in computing statistics. However, the major limitation of DP is that users need to trust the data accumulator. The accumulator gathers actual data from users, and shares the statistics after ensuring the requirements of DP, i.e., no one can identify a user's data with a certain confidence level in the computed aggregate statistics. On the other hand, local differential privacy guarantees privacy of data without involving a trusted accumulator. There exist a number of approaches [9, 10, 15] to ensure LDP for computing histograms and ordinal queries. In [15], the authors developed a solution for protecting LDP of numeric data for computing aggregates (e.g., summation or minimum).

Both DP and LDP assume the same privacy levels for all users, which might not be always the case. In [14], the authors incorporated personalized settings for differential privacy, where a trusted accumulator is required but users can have different privacy levels. Recently, in [3], the authors applied the concept of personalized privacy in the local setting, and developed an approach to ensure personalized LDP (PLDP) of users. However, the approach has limited applicability only for counting queries. In this paper, we develop PENA that guarantees PLDP and can compute any aggregate like average, minimum or maximum.

6 Conclusion

We have developed the first approach, private estimation of numeric aggregates (PENA), to compute aggregates over numeric data while guaranteeing personalized local differential privacy (PLDP). PENA does not involve a central trusted authority and provides users the flexibility to control their privacy levels. Experiments using real datasets show that PENA outperforms modified LDP based approach in terms of accuracy while ensuring higher privacy levels for users for both aggregates average and minimum.

Acknowledgments

This research has been done in the department of Computer Science and Engineering, Bangladesh University of Engineering and Technology (BUET). The work is supported by the research grant from BUET and United International University (UIU).

Bibliography

[1] (2015) Ipums. integrated public use microdata series: Version 6.0. URL https://www.ipums.org/

[2] Bassily R, Smith A (2015) Local, private, efficient protocols for succinct histograms. In: STOC, pp 127–135, DOI 10.1145/2746539.2746632

[3] Chen R, Li H, Qin A, Kasiviswanathan SP, Jin H (2016) Private spatial data aggregation in the local setting. In: ICDE, pp 289–300, DOI 10.1109/ICDE.2016.7498248

[4] Dinur I, Nissim K (2003) Revealing information while preserving privacy. In: PODS, pp 202–210, DOI 10.1145/773153.773173

[5] Dwork C (2006) Differential privacy. In: ICALP, pp 1–12, DOI 10.1007/11787006_1

[6] Dwork C (2011) A firm foundation for private data analysis. Communications of the ACM 54(1):86–95, DOI 10.1145/1866739.1866758

[7] Dwork C, McSherry F, Nissim K, Smith A (2006) Calibrating noise to sensitivity in private data analysis. In: Theory of Cryptography Conference, pp 265–284, DOI 10.1007/11681878_14

[8] Dwork C, Roth A, et al (2014) The algorithmic foundations of differential privacy. Foundations and Trends® in Theoretical Computer Science 9(3–4):211–407, DOI 10.1561/0400000042

[9] Erlingsson Ú, Pihur V, Korolova A (2014) Rappor: Randomized aggregatable privacy-preserving ordinal response. In: CCS, pp 1054–1067, DOI 10.1145/2660267.2660348

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

[11] Hashem T, Kulik L (2007) Safeguarding location privacy in wireless ad-hoc networks. In: UbiComp, pp 372–390, DOI 10.1007/978-3-540-74853-3_22

[12] Hashem T, Kulik L, Zhang R (2010) Privacy preserving group nearest neighbor queries. In: EDBT, pp 489–500, DOI 10.1145/1739041.1739100

[13] Hashem T, Hashem T, Iqbal A (2016) Ensuring feedback data privacy in the context of developing countries. In: ACM DEV, pp 18:1–18:4, DOI 10.1145/3001913.3006627

[14] Jorgensen Z, Yu T, Cormode G (2015) Conservative or liberal? personalized differential privacy. In: ICDE, pp 1023–1034, DOI 10.1109/ICDE.2015.7113353

[15] Nguyên TT, Xiao X, Yang Y, Hui SC, Shin H, Shin J (2016) Collecting and analyzing data from smart device users with local differential privacy. arXiv preprint arXiv:160605053

[16] Rastogi V, Nath S (2010) Differentially private aggregation of distributed time-series with transformation and encryption. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD, DOI 10.1145/1807167.1807247

[17] Sarathy R, Muralidhar K (2011) Evaluating laplace noise addition to satisfy differential privacy for numeric data. Trans Data Privacy 4(1):1–17
[18] Shamir A (1979) How to share a secret. Communications of the ACM 22(11):612–613
[19] Soma SC, Hashem T, Cheema MA, Samrose S (2017) Trip planning queries with location privacy in spatial databases. World Wide Web 20(2):205–236
[20] Sweeney L (2002) Achieving k-anonymity privacy protection using generalization and suppression. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 10(5), DOI 10.1142/S021848850200165X