Continuous Location Statistics Sharing Algorithm with Local Differential Privacy

Fatima Zahra Errounda
Department of Electrical and Computer Engineering
Concordia University
Montreal, Canada
Email: f_errou@encs.concordia.ca

Yan Liu
Department of Electrical and Computer Engineering
Concordia University
Montreal, Canada
Email: yan.liu@concordia.ca

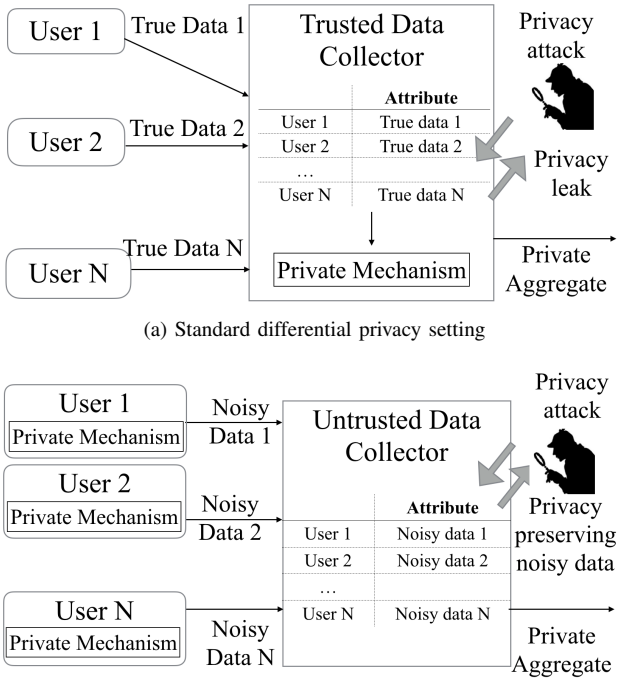
***Abstract*—Continuous sharing of location statistics produces valuable knowledge to understand important phenomena, such as popular places or pattern behaviors. Most importantly, data should be shared without jeopardizing users' privacy. Differential privacy becomes de-facto technique for private statistical data release. Much work focuses on the centralized setting where users send their original data to a trusted server. Then the server adds controlled noises to generate differentially private statistics. This centralized approach is vulnerable to attacks where an adversary may access the true data by attacking the trusted server. Local differential privacy neutralizes this type of attacks by allowing each user to obfuscate their data before it reaches the server for statistical analysis. In this paper, we propose an algorithm to share location statistics that leverages local differential privacy combined with w-event privacy. Our solution guarantees the user's privacy when continuously releasing statistics over infinite streams. Experimental evaluation on real-life data shows our solution with strong privacy guarantee.**

I. Introduction

Continuous sharing of location statistics is used in many application domains, such as traffic estimation [1] and crowd sensing [2] due to the proliferation of data emerging from smart devices and sensors. Data collected from such devices is computed to continuously generate statistics. These statistics are then shared with interested third parties for analysis.

Continuous release of location statistics leads to potential risks of privacy breaches that should not be underestimated. For example, in a real-time traffic estimation, an adversary may reconstruct an individual's trajectory based on the released data, including the number of visits per location or real-time traffic density maps [1]. In fact, an adversary who knows the home and work locations of a given individual may re-identify other location traces even when the individual's identity is not released [3].

Differential privacy [4] becomes de-facto technique for private statistical data release. In a centralized model, a trusted data collector gathers data from different entities, adds carefully calibrated noises, then shares the final result with a data analyst. However, this model is vulnerable to attacks where an attacker may access the original data gathered by the data collector [5]. Local differential privacy is a distributed variant of differential privacy where users obfuscate their personal data themselves, before sending them to the data collector [6]. Hence, it is resilient to the centralized privacy attacks as illustrated in Fig.1.



(a) Standard differential privacy setting

(b) Local differential privacy setting

Fig. 1: Illustration of the local differential privacy resilience to centralized privacy attacks

Much work on local differential privacy focuses on a single data release setting [7], [8], and [9]. However, these works face the challenge of privacy degradation when data is collected over time [10]. Dwork et al. [11] propose two different privacy methods for continuous settings, namely event-level and user-level. The event-level privacy protects a user in a single point in the data stream, however, it does not protect a user in the entire data stream. On the other hand, the user-level privacy requires adding noises in the entire data stream, which reduces the utility of the data in the long run.

w-event privacy [12] is proposed to balance the amount of timestamps that guarantees privacy between the user-level and

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

the event-level privacy. w-event privacy uses a sliding window approach of w timestamps in a data stream within which the user's privacy is guaranteed. When w is set to 1, w-event privacy corresponds to the event-level privacy, when w is set to infinity it converges to the user-level privacy. However, this approach is designed for a centralized differential privacy setting, hence it is vulnerable to the privacy attack shown in Fig.1a. In addition, the statistic similarity is averaged over all users resulting in the over-consumption of the privacy budget by calculating new private releases for users whose statistics are similar.

To solve the problem of centralized privacy attacks and the over-consumption of the privacy budget we propose a solution using the w-event privacy sliding window approach and local differential privacy. Applying local differential privacy in the decision step of w-event privacy raises the challenge of designing a corresponding privacy budget strategy. The **contributions** of this paper are threefold:

1) We leverage local differential privacy to answer the statistics similarity test but locally for each user instead of averaging the statistics difference over all users.

2) We introduce a new privacy management scheme adapted to the randomness induced in the similarity test. We estimate the optimal privacy budget to allocate when a private statistic release is required.

3) We prove that the new privacy budget allocation algorithm satisfies w-event privacy.

We evaluate the utility of our solution using data from a real-world monitoring system. We show that our solution reaches the same level of utility as the state-of-the-art (w-event privacy).

II. Preliminary

We presents the basics of differential privacy with both variants: local differential privacy and w-event privacy.

A. Differential Privacy

Differential privacy is a formal notion of privacy that protects sensitive personal data [4]. It is based on the concept of neighbourhood defined by Dwork [4] as below:

Definition II.1 (Neighbourhood)**.** *Two datasets $\mathcal{D}$ and $\mathcal{D}'$ are neighbours if they differ in at most one element. Meaning that one dataset is a subset of the other and the larger dataset contains exactly one additional row.*

To protect sensitive personal data, differential privacy guarantees that the result of a statistical query on any neighbouring datasets is indistinguishable. Therefore, hindering a strong adversary from inferring private information solely from the query results. Differential privacy utilizes privacy mechanisms to produce answers to statistical queries by adding noise based on a probability distribution to hide a user presence or absence in the dataset. A privacy mechanism $\mathcal{M}$ is a randomized algorithm that outputs a noisy statistics extracted from the dataset $\mathcal{D}$.

Formally, differential privacy is defined as:

Definition II.2 (Differential Privacy)**.** *$\mathcal{M}$ satisfies ϵ-differential privacy if for all neighbouring datasets $\mathcal{D}$ and $\mathcal{D}'$ and for all $\mathcal{O} \in Range(\mathcal{M})$ the following holds:*

$$Pr[\mathcal{M}(\mathcal{D}) \in \mathcal{O}] \leq e^{\epsilon} \times Pr[\mathcal{M}(\mathcal{D}') \in \mathcal{O}]$$

ϵ is called the privacy budget. It controls the degree of indistinguishability between the original and the private statistics, hence the privacy guarantee.

Dwork [13] defines a privacy mechanism to achieve ϵ-differential privacy based on the Laplace distribution. The noise is calibrated with the query's global sensitivity. A query sensitivity is defined as the difference of the query result applied to any two neighbouring datasets. The Laplace mechanism adds noises to the real output using the Laplace distribution as follows:

Definition II.3 (Laplace Mechanism)**.** *For a query q, a mechanism $\mathcal{M}(x) = q(x) + Lap(\frac{\Delta q}{\epsilon})$ satisfies ϵ-differential privacy*

McSherry et al. [14] define the sequential composition mechanism for continuous statistic release as follows:

Theorem II.1 (Sequential Composition)**.** *Consider mechanism $\mathcal{M}_i$ that provides ϵ_i-differential privacy. A sequence of $\mathcal{M}_i$ over a dataset $\mathcal{D}$ provide $\Sigma\epsilon_i$-differential privacy.*

B. Local Differential Privacy

Local differential privacy does not assume that the data collector is trustworthy, therefore, each user adds noise to his original data before sending it to the data collector for statistical analysis. Thus, the original data never leaves the user's environment. Local differential privacy offers the same guarantees as traditional differential privacy and satisfies the composionality Theorem II.1.

The randomized response algorithm is introduced in [13] where the user responds either truthfully or the opposite answer to a sensitive question depending on a coin flip. It is commonly used in surveys of people's "yes or no" opinions about a sensitive question. Because the user gives a randomized response, the surveyors cannot determine the individual's true answer, but can still extract useful statistics [13]. The algorithm is widely adopted by local differential privacy solutions. Although randomized response algorithm is originally designed for binary answer, it has been extended to answer categorical answers [7] and telemetry counts [10].

C. w-event Privacy

w-event privacy [12] is an extension of differential privacy for continuous release of data streams. It redefines the concept of neighbourhood within a sliding window of size w as follows:

Definition II.4 (w-neighbourhood)**.** *Let $S_t = \{D_1, ..., D_t\}$ be a prefix stream of sequential data where at each timestamp i, a dataset D_i is collected with an arbitrary number of rows each corresponding to a unique user. For a positive integer w, two prefix streams S_t and $S_t{}'$ are w-neighbouring if:*

(i) *for each D_i and D_i', such that $i \in [1,t]$ and $D_i \neq D_i'$, it holds that D_i and D_i' are neighbouring, and*

(ii) *for each $i_1 \in [1,t]$ and $i_2 \in [1,t]$, such that $i_1 \neq i_2$ and $i_1 < i_2$, it holds that $i_2 - i_1 + 1 < w$*

A randomization algorithm $\mathcal{M}$ that takes as input a prefix steam is w-event ϵ-differentially private if for any subset $O \in Range(\mathcal{M})$, for all w-neighbouring prefix streams S_t and S_t', and all t, the following holds:

$$Pr[\mathcal{M}(S_t) \in O] \leq e^{\epsilon} \times Pr[\mathcal{M}(S_t') \in O]$$

The privacy mechanism to achieve w-event privacy is composed of two differentially private steps:

The decision step: compares the similarity between the statistics at the current timestamp with the last private release in order to decide if a new private release is required. At each timestamp i, a dataset D_i is built where the rows correspond to individual users, and the columns correspond to the total number of attributes d that a user generates. The statistics similarity is evaluated using the Mean Absolute Error (MAE) as follows:

$$MAE = \frac{1}{d}\sum_{j=1}^{d} \|c_i[j] - c_l'[j]\| \quad (1)$$

Where the vector c_i represents the statistics at timestamp i calculated from the D_i. l is the last timestamp where private statistics are calculated c_l'. To make the decision step differentially private, noise is added to the value of the MAE.

The perturbation step: Perturbation is the process of adding noise to the original statistics. Based on the similarity test result, the perturbation step adds differentially private noise drawn from Laplace distribution to the statistics at timestamp i.

III. Related Work

To provide real-time privacy guarantee in a data stream, several differential privacy are proposed [2], [15], [16]. FAST is a publishing framework based on statistic prediction, statistic estimates, and sampling [15]. The framework releases noisy statistics at sampling points based on the error rate between the estimated and predicted statistics. Although the presented solution optimizes the overall privacy budget management by allocating a portion only at the sample points, it requires the overall length of the data stream for the privacy budget management.

Further work adopts the sampling approach of FAST [15] combined with the sliding window strategy of w-event privacy [2] and [16]. As a result, the sampling points are within a sliding window of w time stamps. The privacy budget is allocated to the sampling timestamps only, while the timestamps in-between are interpolated. All these work achieve a balance between the event-level and the user-level privacy, however, they are not defined in the local differential privacy setting.

RAPPOR (Randomized Aggregatable Privacy-Preserving Ordinal Response) [7] extends the randomized response to other types of inputs such as strings using hash functions. RAPPOR is used in Google Chrome browser to collect the user's data. Kim. et al [17] use RAPPOR to collect the users' locations in an indoor environment. A presence vector of values of "0" and "1" corresponds to the presence or absence of the user at a given known location. All these solutions were designed for the local differential privacy setting. Adopting them to continuous statistics sharing requires designing privacy budget allocation strategies.

IV. Local Differential Privacy Algorithms

The algorithm $\mathcal{M}_i$ at timestamp i is the sequence of the decision algorithm $\mathcal{M}_{1,i}$ and the perturbation algorithm $\mathcal{M}_{2,i}$. The decision algorithm $\mathcal{M}_{1,i}$ is defined in Algorithm 1. It takes the inputs as the current location statistic c_i, the last private release c_r', and the comparison threshold T. It starts by comparing c_i and c_r' with respect to T. Then the randomized response mechanism is executed on the result of the similarity calculation.

Algorithm 1: The decision algorithm $\mathcal{M}_{1,i}$

Input: c_i, c_r', T
Output: Similarity result v_i'
Calculate similarity: $T_i = c_i - c_r'$
$v_i = (T_i > T)$
$v_i' = \begin{cases} 1, & \text{with probability } p \\ 0, & \text{with probability } p \\ v_i & \text{with probability } 1 - 2*p \end{cases}$

The perturbation algorithm $\mathcal{M}_{2,i}$ is defined in Algorithm 2. It takes the inputs as the current location statistic c_i, the last private release c_r', the result of the private similarity test v_i', and the privacy budget consumed so far in the w sliding window $(\epsilon_{i-w+1}, ..., \epsilon_{i-1})$. If the similarity test is positive then the privacy budget for the current timestamp is calculated. Then the Laplace noise is added with the resulting privacy budget. If the similarity test is negative, then the algorithm outputs the last private release as an approximation of the current release without consuming any privacy budget.

Algorithm 2: The perturbation algorithm $\mathcal{M}_{2,i}$

Input: $c_i, c_r', (\epsilon_{i-w+1}, ..., \epsilon_{i-1}), v_i'$
Output: Released count c_i', Allocated budget ϵ_i
if v_i' == *1* **then**
  $\epsilon_i \rightarrow$ Privacy budget allocation strategy
  Add Laplace noise $c_i' \rightarrow c_i + Lap(1/\epsilon_i)$;
else
  $\epsilon_i \rightarrow 0$
  $c_i' \rightarrow c_r'$
end

Given a privacy budget ϵ for a sliding window of size w, we split ϵ in half between the decision step and the perturbation step. We develop a privacy management strategy that allocates

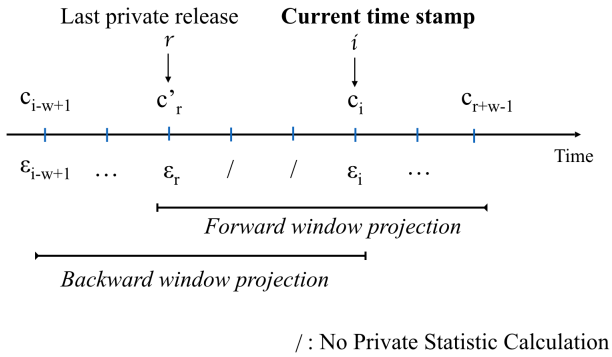


Fig. 2: The window privacy budget for the perturbation step

the privacy budgets ϵ_i at each timestamp as shown in Fig. 2. We estimate the maximum privacy budget to assign to the timestamp i from two perspectives: the backward sliding window projection, and the forward sliding window projection. The backward sliding window begins at timestamp $i-w+1$, and the forward sliding window starts at the last private release r. The two sliding windows are of size w. We calculate the estimated privacy budget from the perspectives of the two windows and select the minimal of the two values to avoid exhausting the privacy budget. Our approach is detailed in Algorithm 3.

Algorithm 3: The privacy budget allocation

Input: $(\epsilon_1, ..., \epsilon_{i-1})$
Output: Allocated budget ϵ_i
Identify the last noisy release c_r
//Forward privacy budget estimation
1 Calculate the remaining timestamps in the forward sliding window projection $k = w - r + i$
2 Calculate the remaining privacy budget for the forward sliding window projection $\epsilon_{remaining} = \epsilon/2 - \epsilon_r$
3 Calculate the estimated forward budget $\epsilon_{i,f} = \epsilon_{remaining}/k$
//Backward privacy budget estimation
4 Calculate the remaining privacy budget for the backward sliding window project $i-w+1$: $\epsilon_{remaining} = \epsilon/2 - \sum_{j=i-w+1}^{i-1} \epsilon_j$
5 Calculate the estimated backward budget $\epsilon_{i,b} = \epsilon_{remaining}/2$
6 Return the minimum of the two estimated budgets $\epsilon_i = min(\epsilon_{i,f}, \epsilon_{i,b})$

V. Privacy Analysis

We first prove that the decision algorithm $\mathcal{M}_1$ and the perturbation algorithm $\mathcal{M}_2$ both satisfy w-event privacy with the privacy budget $\epsilon/2$. We then prove that the overall mechanism $\mathcal{M}$ satisfies w-event privacy with the privacy budget ϵ.

Let $S_t = \{c_1, c_2, ..., c_t\}$ be a data stream, the decision algorithm $\mathcal{M}_1$ on S_t is the composition of the t mechanisms $\mathcal{M}_{1,1}$ to $\mathcal{M}_{1,t}$ where every $\mathcal{M}_{1,i}$ operates on $S_t[i]$ at timestamp i.

Theorem V.1. *The decision algorithm $\mathcal{M}_1$ satisfies w-event privacy with the privacy budget $w \times ln\langle\frac{1-p}{p}\rangle$.*

Proof. Let v_i be the result of comparing the statistic similarity between the current timestamp $S_t[i]$ and the last release. Let v_i' be any output of $\mathcal{M}_{1,i}$ applied to v_i, then:

$$\frac{P(v_i' = 1 \mid v_i = 1)}{P(v_i' = 1 \mid v_i = 0)} = \frac{p + 1 - 2 \times p}{p} = \frac{1-p}{p}$$

Similarly:

$$\frac{P(v_i' = 0 \mid v_i = 0)}{P(v_i' = 0 \mid v_i = 1)} = \frac{1-p}{p}$$

Therefore, $M_{1,i}$ is $ln\langle\frac{1-p}{p}\rangle$-differentially private, because the degree of indistinguishability between the two possible inputs "0" and "1" given an observed output v_i' is $ln\langle\frac{1-p}{p}\rangle$. According to Theorem II.1, for any sliding window of size w, and at any timestamp i in the stream S_t, the privacy budget consumed by $\mathcal{M}_1$ within the sliding window is the sum of the privacy budgets of $\mathcal{M}_{1,k}$ such that $i-w+1 \leq k \leq i$. Therefore the privacy budget of $\mathcal{M}_1$ within a sliding window w is:

$$\epsilon/2 = \sum_{k=i-w+1}^{k=i} ln(\frac{1-p}{p}) = w \times ln\langle\frac{1-p}{p}\rangle$$

where p is $\frac{1}{e^{\frac{\epsilon}{2\times w}}+1}$ Therefore, $\mathcal{M}_1$ satisfies w-event privacy with the privacy budget $\epsilon/2$, which concludes the proof. □

Theorem V.2. *The perturbation algorithm $\mathcal{M}_2$ satisfies w-event privacy with the privacy budget $\epsilon/2$.*

Proof. Let $S_t = \{c_1, c_2, ..., c_t\}$ be a data stream. The perturbation mechanism $\mathcal{M}_2$ is the sequential composition of t mechanisms $\mathcal{M}_{2,1}$ to $\mathcal{M}_{2,t}$, where every $\mathcal{M}_{2,i}$ operates on $S_t[i]$ at timestamp i. To prove that $\mathcal{M}_2$ satisfies w-event privacy with the privacy budget $\epsilon/2$, first we need to prove that each $\mathcal{M}_{2,i}$ is ϵ_i-differentially private. Then, according to Theorem II.1, we need to prove that the sum of the privacy budgets consumed by $\mathcal{M}_2$ within a window of size w does not exceed $\epsilon/2$.

Let $S_t^* = \{c_1', c_2', ..., c_t'\}$ be a data stream such that S_t and S_t' are w-neighbouring. At each timestamp i, the presence or absence of a user changes the counts statistic c_i and c_i' with 1. Therefore, the global sensitivity of the counts statistic is 1. Since we are adding Laplace noise with the scale $1/\epsilon_i$, the mechanism $\mathcal{M}_{2,i}$ is ϵ_i-differentially private as stated in Definition II.3.

Now we prove that for every timestamp i, $\sum_{j=i-w+1}^{j=i} \epsilon_j \leq \epsilon/2$. We assume that private statistics are calculated at each timestamp from $w-i+1$ to i.

$$\sum_{j=i-w+1}^{j=i} \epsilon_j = \sum_{j=i-w+1}^{j=i} min(\epsilon_{j,f}, \epsilon_{j,b})$$
$$\leq min(\underbrace{\sum_{j=i-w+1}^{j=i} \epsilon_{j,f}}_{\text{L}}, \underbrace{\sum_{j=i-w+1}^{j=i} \epsilon_{j,b}}_{\text{R}})$$

We prove that L and R are smaller than $\epsilon/2$. Since $i-w+1$ is the last private release, we replace r with $i-w+1$ and i with $i-w+2$ in Algorithm 3. Then we have:

$$\sum_{j=i-w+1}^{j=i} \epsilon_{j,f} = \epsilon_{i-w+1} + \sum_{i-w+2}^{i} (\epsilon/2 - \epsilon_{i-w+1})/(w+1)$$
$$= \epsilon_{i-w+1} + (w-1) \times (\epsilon/2 - \epsilon_{i-w+1})/(w+1)$$
$$\leq \epsilon_{i-w+1} + (\epsilon/2 - \epsilon_{i-w+1}) = \epsilon/2$$

Therefore $\sum_{j=i-w+1}^{j=i} \epsilon_{j,f} \leq \epsilon/2$.

Similarly, we have:

$$\sum_{j=i-w+1}^{j=i} \epsilon_{j,b} = \epsilon/2 - \sum_{j=i-w+1}^{j=i} \epsilon_{i-w+1}/2^j$$

Without loss of generality, we assume that the maximum privacy budget value $\epsilon/2$ is available at timestamp $i-w+1$, then according Algorithm 3 $\epsilon_{i-w+1} = (\epsilon/2) \times (1/2)$, therefore we have:

$$\sum_{j=i-w+1}^{j=i} \epsilon_{j,b} = \sum_{j=1}^{j=w} (\epsilon/2) \times (1/2^j) = \epsilon/2 \times \sum_{j=1}^{j=w} (1/2^j)$$
$$= \epsilon/2 \times (1 - 1/2^w) \leq \epsilon/2$$

Therefore $\sum_{j=i-w+1}^{j=i} \epsilon_{j,b} \leq \epsilon/2$. We can then conclude:

$$\sum_{j=i-w+1}^{j=i} \epsilon_j \leq min(\sum_{j=i-w+1}^{j=i} \epsilon_{j,f}, \sum_{j=i-w+1}^{j=i} \epsilon_{j,b}) \quad \leq \epsilon/2$$

This concludes the proof. □

Theorem V.3. *$\mathcal{M}$ satisfies w-event privacy with a budget of ϵ.*

Proof. Let S_t be a stream. The sequential composition of the algorithms $\mathcal{M}_1$ and $\mathcal{M}_2$ applied to S_t is the algorithm $\mathcal{M}$. According to the sequential composition Theorem II.1, the privacy budget consumed by $\mathcal{M}$ is the sum of the privacy budgets consumed by $\mathcal{M}_1$ and $\mathcal{M}_2$. As we proved in Theorems V.1 and V.2, $\mathcal{M}_1$ and $\mathcal{M}_2$ both satisfy w-event privacy with budget $\epsilon/2$ respectively. Therefore, $\mathcal{M}$ satisfies w-event privacy with a budget of ϵ. □

VI. Experimental Evaluation

We implement the budget absorption and the budget distribution algorithms of w-event privacy and compared them with our budget allocation algorithm.

A. Dataset

We connect to an actual online proximity detection monitoring deployed by a telecommunication company in Montreal. Three sensors were deployed in downtown Montreal that generate streams of information about devices within their connectivity range using WiFi as depicted in Fig. 3. An example of the data collected is illustrated below:

'DateTime':2018-07-09T12:00:00+00:00, 'rssi':0, 'frequency':5180, 'address':0a:0b:0c:8d:ff:cf

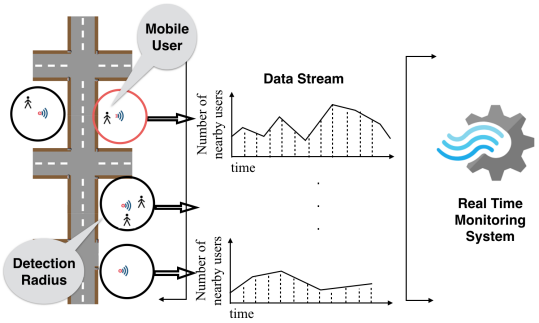


Fig. 3: Real-time proximity detection scenario

The "DateTime" is the exact time at which the user was detected by the detection device, the "rssi" is the strength of the radio signal coming out of the user's smart mobile device, the "frequency" is the radio frequency used by the user's smart mobile device for communication, and the "address" is the MAC address of the smart mobile device.

B. Experiment Setup

The three sensors send the collected data in real time to Azure IoT hub. We retrieve data from the IoT hub resulting in 6054 records captured at approximately every 1.6 second. We do not run the privacy budget algorithms on the sensors because we first need to prove our concept, instead we run on a machine with AMD Core 4 CPU 2.5GHz and 12GB RAM, running Windows 10 using Python 3.6.5.

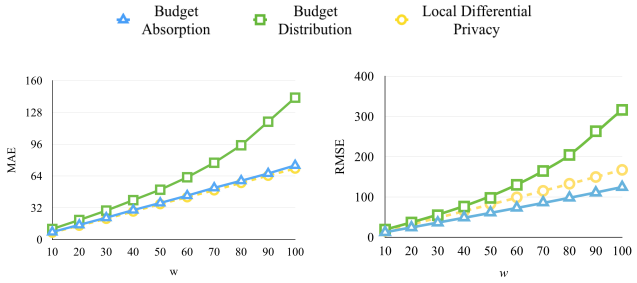
a) Metrics: We evaluate the utility of the three implemented methods using the Mean Absolute Error and the Root Mean Square Error of the perturbation error. The two metrics are described in Equations (1) and (2).

$$RMSE = \sqrt{\frac{1}{d} \sum_{j=1}^{d} (x[j] - y[j])^2} \tag{2}$$

Both metrics are standard for measuring the magnitude of change of a varying quantity. We measure the results of the state-of-the-art budget management strategies and our solution while fixing the privacy budget and varying the window size. The budget distribution strategy allocates the privacy budget in an exponentially decreasing fashion, favouring the earlier timestamps with exponentially more budget than the later ones. Initially, the budget absorption strategy distributes the budget uniformly to all w timestamps, then the accumulated privacy budget from timestamps where a noisy release is not required becomes available for future private release. The goal is to compare the utility of our solution with the state-of-the-art strategies.

C. Evaluation Results

a) Varying w: Fig. 4b investigates how utility of the three budget allocation algorithms changes with different values of w. We fixed the privacy budget ϵ to 1. We observe that the mean absolute error is almost the same for our solution and the budget absorption algorithm. This is because we also



(b) MAE (c) RMSE

Fig. 4: The perturbation error vs w

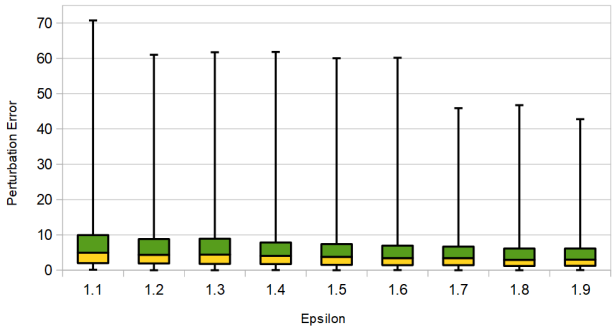


Fig. 5: The perturbation error vs ϵ

adopt a uniform privacy budget allocation just like the budget absorption algorithm. However, we cannot cancel future releases like the privacy absorption strategy because of the unpredictability of the randomized response similarity test. We solve this issue by bounding the privacy budget used by the future timestamps with the remaining privacy budget from the backward projection of the sliding window.

The root mean squared error of our solution is slightly higher than the budget absorption strategy as shown in Fig. 4c. This is because our solution may result in more private releases on a data stream. The total number of releases in a stream in both solutions relies on testing the similarity of the statistics at each timestamp. However, the budget absorption algorithm cancels some private releases to absorb the privacy budget, while the total number of releases in our solution depends on the randomized response parameter f that does not change considerably with the increase of the sliding window size w.

b) Varying ϵ: We measure the perturbation error as the difference between the true and the noisy counts released by our mechanism. To evaluate the impact of the privacy budget change on data utility, we fix the window size to w=4, then we vary the privacy budget ϵ between 1.1 and 1.9. We build a box plot based on the perturbation error. As illustrated in Fig.5, the medians do not change significantly for different values of ϵ showing that our solution can provide practical utility with strong privacy guarantee.

VII. Conclusion and Future Work

The goal of this work is to explore the use of local differential privacy in continuous sharing of location statistics. We define a new privacy budget allocation algorithm for the w-event privacy to guarantee the user's privacy in continuous statistic sharing. We achieve the same utility as the centralized approach (w-event privacy) with longer real-life data streams (1 hour vs 10 minutes for w-event privacy experimental evaluation). Future work includes testing our approach with longer streams, and computing optimization to run our solution on IoT devices.

References

[1] J. L. Ny, A. Touati, and G. J. Pappas, "Real-time privacy-preserving model-based estimation of traffic flows," in *Cyber-Physical Systems (ICCPS), 2014 ACM/IEEE International Conference on*, April 2014, pp. 92–102.

[2] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy," *IEEE Transactions on Dependable and Secure Computing*, 2016.

[3] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM, 2011, pp. 145–156.

[4] C. Dwork, "Differential privacy: A survey of results," in *International Conference on Theory and Applications of Models of Computation*. Springer, 2008, pp. 1–19.

[5] T. Murakami and Y. Kawamoto, "Restricted local differential privacy for distribution estimation with high data utility," *arXiv preprint arXiv:1807.11317*, 2018.

[6] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*. IEEE, 2013, pp. 429–438.

[7] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*. ACM, 2014, pp. 1054–1067.

[8] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and S. Y. Philip, "Lopub: High-dimensional crowdsourced data publication with local differential privacy," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, 2018.

[9] T. Wang, N. Li, and S. Jha, "Locally differentially private frequent itemset mining," in *Locally Differentially Private Frequent Itemset Mining*. IEEE, 2018, p. 0.

[10] B. Ding, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Advances in Neural Information Processing Systems*, 2017, pp. 3571–3580.

[11] C. Dwork, M. Naor, T. Pitassi, and G. N. Rothblum, "Differential privacy under continual observation," in *Proceedings of the forty-second ACM symposium on Theory of computing*. ACM, 2010, pp. 715–724.

[12] G. Kellaris, S. Papadopoulos, X. Xiao, and D. Papadias, "Differentially private event sequences over infinite streams," *Proceedings of the VLDB Endowment*, vol. 7, no. 12, pp. 1155–1166, 2014.

[13] C. Dwork, A. Roth *et al.*, "The algorithmic foundations of differential privacy," *Foundations and Trends® in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

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

[15] L. Fan, L. Xiong, and V. Sunderam, "Fast: differentially private real-time aggregate monitor with filtering and adaptive sampling," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*. ACM, 2013, pp. 1065–1068.

[16] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Rescuedp: Real-time spatio-temporal crowd-sourced data publishing with differential privacy," in *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, April 2016, pp. 1–9.

[17] J. W. Kim, D.-H. Kim, and B. Jang, "Application of local differential privacy to collection of indoor positioning data," *IEEE Access*, 2018.