Private Continual Release of Real-Valued Data Streams

Victor Perrier
ISAE-SUPAERO
& Data61, CSIRO
v.perrier@gmail.com

Hassan Jameel Asghar
Macquarie University
& Data61, CSIRO
hassan.asghar@mq.edu.au

Dali Kaafar
Macquarie University
& Data61, CSIRO
dali.kaafar@mq.edu.au

***Abstract*—We present a differentially private mechanism to display statistics (e.g., the moving average) of a stream of real valued observations where the bound on each observation is either too conservative or unknown in advance. This is particularly relevant to scenarios of real-time data monitoring and reporting, e.g., energy data through smart meters. Our focus is on real-world data streams whose distribution is *light-tailed*, meaning that the tail approaches zero at least as fast as the exponential distribution. For such data streams, individual observations are expected to be concentrated below an unknown threshold. Estimating this threshold from the data can potentially violate privacy as it would reveal particular events tied to individuals [1]. On the other hand an overly conservative threshold may impact accuracy by adding more noise than necessary. We construct a utility optimizing differentially private mechanism to release this threshold based on the input stream. Our main advantage over the state-of-the-art algorithms is that the resulting noise added to each observation of the stream is scaled to the threshold instead of a possibly much larger bound; resulting in considerable gain in utility when the difference is significant. Using two real-world datasets, we demonstrate that our mechanism, on average, improves the utility by a factor of 3.5 on the first dataset, and 9 on the other. While our main focus is on continual release of statistics, our mechanism for releasing the threshold can be used in various other applications where a (privacy-preserving) measure of the scale of the input distribution is required.**

I. Introduction

Many services can benefit from real-time monitoring of statistics from customer data. Examples include electricity usage in a neighbourhood collected through smart meters, customers' expenditure in a supermarket on a given day, and commute time of residents of a city during peak hours. Statistics for these applications can be obtained from real-time data collected through a variety of sensors and refreshed as new data arrives. These statistics can then be displayed to analysts and planners who could use them to optimize services. Privacy concerns, however, preclude release of raw statistics. For instance, a customer at a pharmacy would not be willing to disclose the purchase of medicines linked to a peculiar health condition. Likewise, analysis of smart meter data can likely reveal the activities of a particular household or even whether anyone is at home or not. Such privacy violations have been demonstrated for the case of smart meter data where patterns such as the number of people in the household as well as sleeping and eating routines were revealed even without any prior training [1]. The goal therefore is to enable monitoring of statistics without compromising individual privacy.

A natural candidate for privacy protection is the rigorous framework of differential privacy [2], [3]. Informally, any algorithm satisfying the definition of differential privacy has the property that its output distribution (based on the coin tosses of the algorithm) on a given database is close in probability to the output distribution if any single row in the dataset is replaced. The closeness is parameterized by the privacy budget ϵ. Most of the work on differential privacy has focused on static (input) datasets, and there has been very little focus on datasets that are continuously being updated as in our setting [4], [5]. Despite this, there is a growing need to shift focus to provide privacy in the dynamic setting which is likely to be more pervasive in the near future [6].

More precisely, our scenario is concerned with releasing statistics from a sequence of observations arriving in a streaming fashion each within some public upper bound B. Our statistic of interest is the continually changing average as new observations arrive. This can be readily obtained by summing all the observations seen thus far (since the number of observations is assumed public). We remark that our focus is on approaches that provide *event level* privacy [4] only, which means that individuals are guaranteed that their peculiar events remain private but not necessarily the general trend.[1] For many use cases this is a suitable guarantee of privacy, e.g., individuals might be happy to disclose their routine trip to work while unwilling to share the occasional detour. One way to release the sum via differential privacy is to add independent noise generated through the Laplace distribution scaled to B [2]. However, this results in cumulative error (absolute difference from the true sum) of $O(B\sqrt{n})$ after n observations. Two aforementioned works on continual release of datasets, i.e., [4] and [5], focus on binary streams, where each observation is either 0 or 1. We can generalize their algorithm to observations within the bound B which results in a considerably reduced error of $O(B(\log_2 n)^{1.5})$.

While this significantly reduces the error over the basic approach, the error is still proportional to B. In many real

[1] The latter is guaranteed through *user level* privacy, i.e., privacy for all events from a user. See [3, §12] and [4] for a further discussion on the merits of event versus user-level privacy.

Network and Distributed Systems Security (NDSS) Symposium 2019
24-27 February 2019, San Diego, CA, USA
ISBN 1-891562-55-X
https://dx.doi.org/10.14722/ndss.2019.23535
www.ndss-symposium.org

world situations, the bound B might not be known in advance, or known only as the worse case bound resulting in an overly conservative estimate of the true bound. Likewise, perhaps most observations are tightly concentrated below an unknown threshold τ well below B. For instance, returning to our commute time use case, it is highly unlikely that anyone would be commuting for the full 24 hours on a given day. We are interested in a mechanism that allows us to determine a threshold τ below which majority of the observations are concentrated. This in turn allows to release statistics with noise scaled to τ rather than B resulting in error $O(\tau(\log_2 n)^{1.5})$, which is a significant improvement depending on B, τ and n.[2]

However, estimating τ is not straightforward due to a number of reasons. First, estimating τ beforehand would result in high or even unbounded cumulative error due to outliers. Thus, any algorithm needs to observe at least a small subset of initial observations before determining τ. This *time lag* needs to be optimised for accuracy: estimating τ too early will result in high accumulated error, and too late will only show marginal improvement over the default case (i.e., when using B as the estimate). Likewise, again for reasons of accuracy, we need to ensure that readings outside the threshold are sporadic. Finally and most importantly, naively estimating τ can result in privacy violation by leaking information specific to an individual, e.g., if we take the maximum of the observations seen so far as τ, we display the exact value corresponding to a particular event from an individual.

In this paper, we propose a mechanism that allows us to estimate the threshold τ using a subset of observations from an incoming stream via differential privacy, simultaneously optimizing utility for releasing the moving average. Although we optimize utility for the case of moving averages, our mechanism for releasing the threshold is generic enough to be used for other statistics and applications. These include displaying the average with a sliding window [7] or releasing histogram of the streaming data [8] where in all cases the noise will be scaled to the most concentrated part of the distribution of the stream.

In addition to theoretical accuracy guarantees, we provide empirical evidence of the utility gain of our scheme using two real world datasets: the first dataset contains about 50 million individual trip times on public trains in the city of Sydney (Australia) over a period of two weeks, and the second dataset is composed of individual amount spent over 140,000 transaction by about 1,000 customers in a major Australian supermarket. Using the two datasets we first verify that real world data has the property that most readings are concentrated tightly well below a conceivable conservative bound B. Using the same datasets we then show that our improved algorithm displays the average statistic (commute time or amount spent) with a utility many orders of magnitude (≈ 3.5 and 9 resp., on the two datasets) better than applying (generalized versions of) the state of the art algorithms [4], [5]. Our utility gain is for data streams that obey a *light-tailed distribution*, namely a distribution whose tail lies below the exponential distribution (beyond the above mentioned threshold; see Section II-C for a precise definition). We argue and show that many real-world datasets are expected to satisfy this property.[3]

II. Background

In this section we formally describe our problem, associated definitions and overview of the algorithm from [4] and [5] referred to as the binary tree (BT) algorithm which will serve both as a benchmark and a sub-module of our technique.

A. Problem Statement

Let B be a positive real number. We model input streams (or strings), denoted σ, as the set of finite strings $\Sigma = [0, B]^{\mathbb{N}}$ of length at most n. The ith element of σ shall be denoted by $\sigma(i)$, and shall be called the ith observation or reading. A generic element or observation from σ shall be denoted by x. For $j \geq i$, $\sigma(i{:}j)$ represents the substring (or sub-stream) $\sigma(i)||\cdots||\sigma(j)$, where $||$ is the concatenation operator. We are interested in finding the average of the elements of the stream σ at each time step $i \in \mathbb{N}$. This reduces to finding $\sum_{j=1}^{i} \sigma(j)$ at each step $i \in [n]$, since we assume the *observation counter* to be public. Our goal is to release a privacy-preserving version of this sum.

B. Privacy Definitions

Definition 1 (Sum Query)**.** We call the function $c : \Sigma \times \mathbb{N} \to \mathbb{R}$ defined for $\sigma \in \Sigma$ and $i \in [n]$ as $c(\sigma, i) = \sum_{j=1}^{i} \sigma(i)$ as the sum query.

Definition 2 (Adjacent Streams)**.** Let $\sigma, \sigma' \in \Sigma$, The Hamming distance $d(\sigma, \sigma')$ is the number of elements different in the corresponding positions of the two strings, i.e., $d(\sigma, \sigma') = |\{i : \sigma(i) \neq \sigma'(i), \forall i \in \mathbb{N}\}|$. The two streams σ and σ' are adjacent if and only if $d(\sigma, \sigma') = 1$.

Definition 3 ((ϵ, δ)-Differential Privacy)**.** A summation mechanism M is (ϵ, δ)-differentially private if and only if for any two adjacent streams σ, σ' we have $\forall n \in \mathbb{N}$ and $\forall S \subset \mathbb{R}$,

$$\Pr\left[M(c, \sigma, n) \in S\right] \leq \Pr\left[M(c, \sigma', n) \in S\right] \times e^{\epsilon} + \delta,$$

where ϵ is a small constant and δ is a negligible function in n. We shall use $\hat{c}$ to denote the output of M in the following.

Note that the notion of differential privacy for streaming data is the same as that for static datasets. The difference lies in how neighboring datasets are defined. In the case of streaming data, neighboring datasets are defined as streams differing in one element (one event anywhere in the stream). The privacy definition does not assume the stream σ to have any specific distribution, barring the fact that each of its element is within $[0, B]$. For utility however we shall assume that the streams are sampled with some underlying probability distribution with support over the set $[0, B]$.

Definition 4 (Probability Distribution of Streams)**.** Let $B \in \mathbb{R}^+$. Denote by $\mathcal{F}_B$ the probability distribution which satisfies

[2] For instance, assume $n = 1{,}000{,}000$ and the known bound is $B = 10{,}000$, and we are interested in the average. Assume further that almost all observations are within $\tau = 100$ with an average of 30. Then, through the original mechanism we get the (noisy) average as 30 ± 1. Through the mechanism that scales noise according to τ, we get the noisy average as 30 ± 0.01, an improvement by a factor of $B/\tau = 100$. This can be significant if the average is required with high precision.

[3] Also see our discussion on what real-world datasets are likely to be light-tailed versus heavy-tailed in Section IX.

$\Pr[X \in [0, B]] = 1$, for any random variable X distributed as $\mathcal{F}_B$. A string σ is said to have distribution $\mathcal{F}_B$, if for all $i \in \mathbb{N}$, $X_i = \sigma(i)$ is sampled from $\mathcal{F}_B$. We denote this by $\sigma \leftarrow_{\mathcal{F}_B} \Sigma$.

Definition 5 ((α, β) Utility)**.** The mechanism $\hat{c}$ is said to be (α, β)-useful if for all $n \in \mathbb{N}$ and $\sigma \leftarrow_{\mathcal{F}_B} \Sigma$,

$$\Pr[|\hat{c}(\sigma, n) - c(\sigma, n)| \leq \alpha] \geq 1 - \beta,$$

where the probability is over the coin tosses of $\hat{c}$ and the distribution $\mathcal{F}_B$.

Note that the above is different from the utility definition in [5], where the probability is over the coin tosses of $\hat{c}$ only, and hence the inequality is satisfied for all strings σ. In our case, we shall be utilizing the probability that certain strings are more likely realized in practice; hence the use of the distribution $\mathcal{F}_B$. We stress again that the privacy definition does not rely on $\mathcal{F}_B$.

Consider an arbitrary function $c : \Sigma \to \mathbb{R}$. The sum query falls under this definition with an auxiliary parameter $n \in \mathbb{N}$. We first define the global sensitivity of c.

Definition 6 (Global Sensitivity)**.** The global sensitivity of a function $c : \Sigma \to \mathbb{R}$, denoted GS, is defined as

$$\mathsf{GS}(c) = \max_{\sigma, \sigma' \in \Sigma \,:\, d(\sigma, \sigma') \leq 1} |c(\sigma) - c(\sigma')|.$$

Definition 7 (Laplace Mechanism)**.** Let $\text{Lap}(b)$ denote the probability density function of the Laplace distribution with mean 0 and scale b given as $\text{Lap}(b) = \frac{1}{2b} \exp\left(\frac{|x|}{b}\right)$. Then the mechanism $\hat{c}(\sigma) = c(\sigma) + Y$, where Y is drawn from $\text{Lap}(\frac{\mathsf{GS}}{\epsilon})$ is $(\epsilon, 0)$-differentially private [2].

A definition of sensitivity that is defined for a particular input string σ is called local sensitivity.

Definition 8 (Local Sensitivity)**.** The local sensitivity of a function $c : \Sigma \to \mathbb{R}$ at $\sigma \in \Sigma$, denoted LS_σ, is defined as

$$\mathsf{LS}_\sigma(c) = \max_{\sigma' \in \Sigma \,:\, d(\sigma, \sigma') = 1} |c(\sigma) - c(\sigma')|.$$

The advantage of using local sensitivity is that we only need to consider neighboring strings of σ which could result in lower sensitivity of the function c, and consequently lower noise added to the true answer c. Unfortunately, replacing the global sensitivity with local sensitivity naively in the Laplace mechanism (for instance) may not result in differential privacy [9]. This drawback can be removed by using smooth sensitivity [10] instead.

Definition 9 (Smooth Upper Bound)**.** For $b > 0$, an b-smooth upper bound on LS_σ, denoted SS^*_σ satisfies:

$$\mathsf{SS}^*_\sigma(c) \geq \mathsf{LS}_\sigma(c), \ \forall \sigma \in \Sigma,$$
$$\mathsf{SS}^*_\sigma(c) \leq e^b \mathsf{SS}^*_{\sigma'}(c), \ \forall \sigma, \sigma' \in \Sigma : d(\sigma, \sigma') = 1.$$

Definition 10 (Smooth sensitivity)**.** For $b > 0$, the b-smooth sensitivity of c, denoted $\mathsf{SS}_{\sigma,b}(c)$, at $\sigma \in \Sigma$ is

$$\mathsf{SS}_{\sigma,b}(c) = \max_{\sigma' \in \Sigma} \left\{ \mathsf{LS}_{\sigma'}(c) \cdot e^{-bd(\sigma, \sigma')} \right\}.$$

Note that smooth sensitivity is the smallest function to satisfy the definition of a smooth upper bound [10]. Smooth sensitivity allows us to add noise proportional to $\frac{\mathsf{SS}_{\sigma,b}}{a}$ to the output of the function c to obtain (ϵ, δ)-differential privacy. The choice of a and b depends on the privacy parameters and the distribution used to generate noise [10].

C. Statistical Definitions

Definition 11 (p-Quantile)**.** Let F be a cumulative distribution function (CDF) of some continuous random variable X. The p-quantile of F, denoted x_p, is defined as

$$x_p = \inf\{x \in \mathbb{R} : F(x_p) = \Pr(X \leq x_p) \geq 1 - p\}.$$

Fact 1. *Let X be an exponentially distributed random variable. Then its CDF is given by*

$$H(x; \gamma) = \begin{cases} 1 - e^{-\gamma x}, & x \geq 0, \\ 0, & x < 0. \end{cases}$$

Let $0 \leq p < 1$. The quantile function of H is given as

$$H^{-1}(p; \gamma) = -\frac{\ln p}{\gamma}. \tag{1}$$

Definition 12 (Light-tailed distribution)**.** Let X be a random variable with CDF F and let Y be an exponentially distributed random variable with CDF $H(\cdot; \gamma)$. Let x_p be the p-quantile of F. Let $\gamma = -\frac{\ln p}{x_p}$, so that the p-quantile of H, i.e., y_p, is equal to x_p. We say that X has a *light-tailed distribution beyond* x_p, or equivalently F is light-tailed beyond x_p, if $\forall x \geq x_p$, $F(x) \geq H(x; \gamma)$.

The choice $\gamma = -\frac{\ln p}{x_p}$ is immediate from Eq. 1.

Proposition 1. *Let X be exponentially distributed with CDF $H(\cdot; \gamma)$. Let $r \geq 1$. Let x_p be the p-quantile of H and let x_{p^r} be the p^r-quantile of H. Then*

$$x_p \cdot r \geq x_{p^r}. \tag{2}$$

Proof: When $r = 1$, we trivially have $x_{p^r} = x_p = x_p \cdot 1$. So, consider $r > 1$ and assume to the contrary that $x_p \cdot r < x_{p^r}$. From Eq. 1, this implies that

$$\begin{aligned} -\frac{\ln p^r}{\gamma} &< -\frac{\ln(pr)}{\gamma} \\ \Rightarrow -\ln p^r &< -\ln(pr) \\ \Rightarrow -r \ln p &< -\ln(pr) < -\ln p, \end{aligned}$$

which implies $r < 1$, a contradiction. ■

Proposition 2. *Let X be a random variable with CDF F. Let x_p be the p-quantile of F. The expected number of samples required to observe at least a constant number of samples $x \in X$ such that $x \geq x_p$ is $\Omega(\frac{1}{p})$.*

Proof: This follows from the properties of the binomial distribution. The probability that a sample $x \in X$ satisfies $x \geq x_p$ is given by p. In m samples, we expect mp successes. Setting $mp \geq c$ for some constant c gives us $m = \Omega(\frac{1}{p})$. ■

D. The Binary Tree (BT) Algorithm

To release the private version of the sum $c(\sigma, i)$ at step $i \in [n]$, we shall use the binary tree algorithm from [4], [5] as a building block. We call this the BT algorithm. We briefly outline the algorithm, and discuss what we would like to improve. Given a string of length n, the BT algorithm first constructs a complete binary tree: the leaves are labelled by the *integer* intervals[4] $[1..1], [2..2], \dots, [n..n]$ and each parent node is the union of intervals of its two child nodes. To output the noisy count $\hat{c}(\sigma, i)$, the algorithm finds at most $\log_2 n$ nodes in the binary tree, whose union equal $[1..i]$. Thus, instead of adding noise of scale $\frac{Bn}{\epsilon}$ (for ϵ-differential privacy through a simple application of the Laplace mechanism), the noise added to each node is only scaled to $\frac{B \log_2 n}{\epsilon}$, resulting in an ϵ-differentially private algorithm. For more concrete details, see [4], [5].

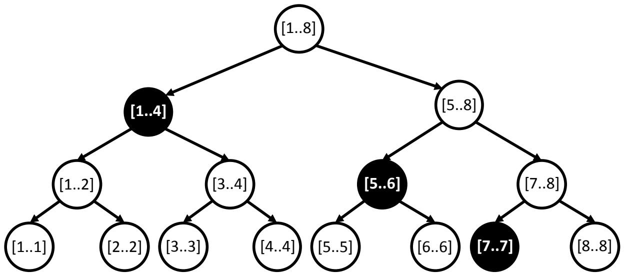


Fig. 1. Example of the binary tree algorithm [4], [5]. To compute the sum of the first 7 observations, noise is added to only 3 nodes whose union equals the interval $[1..7]$.

Figure 1 illustrates the algorithm with an example. We have $n = 8$, and we wish to find $\hat{c}(\sigma, 7)$. This can be done by adding only three noisy sums corresponding to the nodes $[1..4]$, $[5..6]$ and $[7..7]$ (shaded in the figure). That is

$$\hat{c}(\sigma, 7) = \sum_{i=1}^{4} \sigma(i) + \text{Lap}\left(\frac{B \log_2 8}{\epsilon}\right) + \sum_{i=5}^{6} \sigma(i) + \text{Lap}\left(\frac{B \log_2 8}{\epsilon}\right) + \sigma(7) + \text{Lap}\left(\frac{B \log_2 8}{\epsilon}\right).$$

E. Goal

If the stream σ has distribution $\mathcal{F}_B$ (see Definition 4), then the global sensitivity of the function c is $\mathsf{GS} = B$. For strings of length n, the BT algorithm has error [5]

$$\alpha = O\left(\mathsf{GS} \cdot \frac{1}{\epsilon} \sqrt{8 \ln \frac{1}{\beta}} (\log_2 n)^{1.5}\right), \tag{3}$$

with probability at most β. Since $\mathsf{GS} = B$, we get a linear term in B. We aim to improve the dependence on B. Our gain is on input streams σ whose distribution is light-tailed beyond a threshold $\tau \ll B$ (see Definition 12). In other words, input streams whose distribution is concentrated below τ. Then, instead of using global sensitivity, we will use smooth sensitivity tailored to the threshold τ. This means that the noise added will be proportional to τ rather than B in the BT algorithm, resulting in improved utility. In the next section, we give several examples of real-world datasets which are light-tailed. Thus, our improved approach has practical utility gains.

III. Motivation and Overview of the Proposed Mechanism

A. Motivation

There are many scenarios where an upper bound B on a generic element of the stream is overly conservative:

- There might not be a natural bound B known in advance, e.g., a bound on the expenditure during a trip to the supermarket. Any guess on the bound B would be taking into account instances of unusually high spendings. This will result in a very conservative upper bound.
- In some cases, a natural bound B may exist. For instance, public transport commute time per day has a natural bound of $B = 24$ hours. However, most commute times will be tightly concentrated well below this B. Once again, we would have an overly conservative estimate.

Thus, our aim is to obtain a more realistic threshold $\tau \ll B$ tailored to the input stream.

B. Empirical Validation of Concentration of Data

Here we validate our assumption that in many cases real data is concentrated well below a conceivable bound B. We consider two real world datasets:

- *Train trips dataset:* This consists of commute times of trips made by passengers through public trains in the greater region surrounding the city of Sydney in Australia.[5] The aim here is to display the average commute time in real-time (using the cumulative sum). Informally, the privacy issue here is to hide the exact travel time of any single trip of an individual, as it may lead to inferring the individual's exact location at a given time. The total number of train trips in the dataset is about 50 million (spanning over 4 weeks).
- *Supermarket expenditure dataset:* A dataset showing the amount of money spent by customers of a major supermarket retailer in Australia. The goal is to show real-time average expenditure. Informally, privacy property here is to hide the exact transaction amount of a customer on a trip to the supermarket, which may disclose the type of products bought by the customer. This dataset is much smaller and contains about 140,000 transactions (a transaction contains multiple purchased items) by approximately 1,000 customers over a period of one year.

The distribution of both datasets, when viewed as input streams, satisfies our definition of a light-tailed distribution (cf. Definition 12). This is depicted in Figure 2. The left graph shows the (smoothed) empirical cumulative distribution function (ECDF) of the time taken by a trip by a passenger

[4] For real numbers a and b, such that $a \leq b$, the notation $[a..b]$ denotes the set of integers $\mathbb{Z} \cap [a, b]$.

[5] The commute time per trip also includes any waiting time after and before a passenger has tapped on and off through the smart ticketing system.

in the train trips dataset. We can see that the peak is around 20 to 30 mins, and very few customers take more than 150 minutes on a given day. Notice that this is significantly less than the maximum possible time in a 24 hour period (i.e., 1,440 minutes). Likewise, we see a similar trend in the total expenditure during a trip to the supermarket on a given day, shown in Figure 2 (right). Many other similar datasets are expected to have a light tailed distribution, e.g., phone-call durations or smart electricity meter readings. See Section IX for a detailed discussion on why such datasets are likely to exhibit a light-tailed distribution.

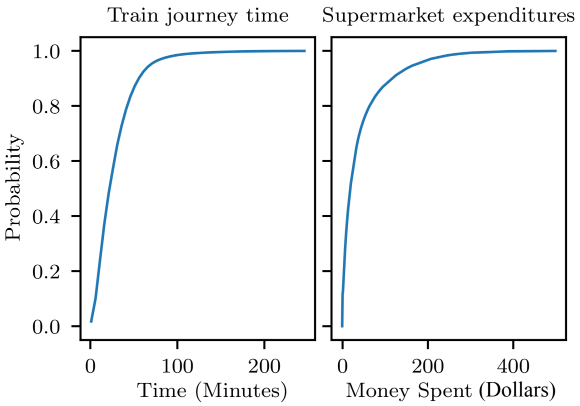


Fig. 2. ECDF of journey times (in minutes) from the train trips dataset (left) and expenditures (in dollars) from the supermarket dataset (right). Both are concentrated well below any conservative bound.

C. Overview of the Mechanism

Figure 3 shows the pictorial overview of our approach. We will first (privately) estimate a threshold τ using the first $m < n$ observations. We then release the sum of the first m observations, i.e., $c(\sigma, m)$, at once using an application of the simple Laplace mechanism with noise scaled to $\approx \tau$. For each step after m, we continually release the sum $c(\sigma, i)$ for $m < i \leq n$ using the BT algorithm with Laplace noise once again scaled to $\approx \tau$. As a consequence, we withhold releasing the sum before we have enough data points, quantified by m, from the stream to estimate τ. We shall call m the *time lag*. Globally, there are two sources of error that we seek to minimise. First is the *outlier error* (denoted α_{out}): any readings above τ will be stripped to τ (before adding noise). This error can occur with probability β_{out} indicated in the figure. The second is the accumulated error at the last step, i.e., n, due to Laplace noise whose probability is denoted β_{Lap} in the figure.

Privacy

The overall mechanism is outlined in Algorithm 1. The mechanism is (ϵ, δ)-differentially private, where $\epsilon = \epsilon_1 + \epsilon_2$. Steps 1 and 2 are altogether $(\epsilon_1 + \epsilon_2, \delta + 0) = (\epsilon, \delta)$-differentially private due to the basic composition property of differential privacy [3]. Steps 3 to 6 are $(\epsilon, 0)$-differentially private due to the properties of the BT algorithm. Since the two sub-streams $\sigma(1{:}m)$ and $\sigma(m+1{:}n)$ are disjoint, overall we have (ϵ, δ)-differential privacy due to the parallel composition property of differential privacy [11]. Notice that while we release the sum of the first m observations in one step, the mechanism can be modified to release the sum of each of the first m observations through the BT algorithm, if required (with noise scaled as $\approx \tau \log_2(m)/\epsilon_1$).

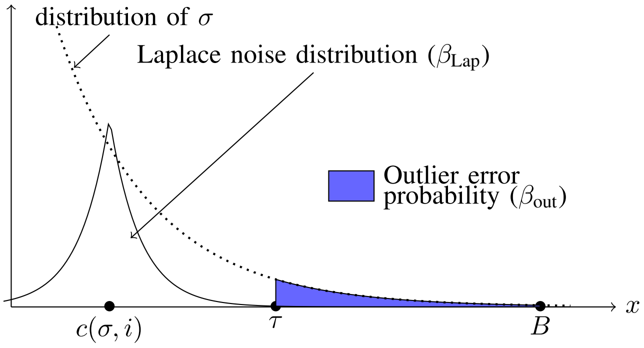

distribution of σ
Laplace noise distribution (β_{Lap})
Outlier error probability (β_{out})
$c(\sigma, i)$
τ
B
x

Fig. 3. Conceptual diagram of our approach. Once the threshold τ has been determined, there are two sources of errors: error due to outliers and error due to the Laplace noise added to the sum $c(\sigma)$.

```
Mechanism 1: Proposed Global Mechanism
input: Input stream $\sigma$, stream length $n$, time lag
       $m \leq n$, privacy parameters $\epsilon$ (split between $\epsilon_1$
       and $\epsilon_2$) and $\delta$.
Estimate $\tau < B$ based on the first $m$ values of $\sigma$
  through the mechanism described in Section IV giving
  us an $(\epsilon_1, \delta)$-differentially private algorithm.
Release $\hat{c}(\sigma, m)$ with the Laplace mechanism with noise
  scaled to $\frac{\tau}{\epsilon_2}$.
for $i = m + 1$ to $n$ do
    if $\sigma(i) > \tau$ then
        Set $\sigma(i) \leftarrow \tau$.
    Use the BT algorithm with noise scaled to
      $\approx \tau \log_2(n - m)/\epsilon$ to release $\hat{c}(\sigma, i)$.
```

IV. Privately Estimating the Threshold τ

In this section, we will find how to estimate and then privately release the threshold τ. Ideally, τ should simultaneously minimize the time lag m and the outlier error α_{out} (characterized by the probability β_{out}). We discarded several straightforward ways of privately computing τ. For instance,

- The most obvious choice is the maximum of the m values. To make this differentially private, we need to scale noise according to the sensitivity of the max function. If we use global sensitivity, the estimated threshold τ will be approximately B, resulting in no utility gain. We could instead use smooth sensitivity [10], but since a possible neighbour of the target stream σ may have any value between 0 and B, this would again result in sensitivity close to B.
- Another alternative is to use the standard deviation of the underlying input distribution $\mathcal{F}_B$ of σ. However, this requires knowing the distribution in advance. We are interested in a more general problem where only a few simple assumptions about the distribution $\mathcal{F}_B$ hold true and are known beforehand.

Our statistic of choice is the p-quantile (cf. Definition 11). This can be privately computed using an algorithm similar to the algorithm for computing the median of a sequence using smooth sensitivity [10]. Analogous to Definition 11, the p-quantile of a stream $\sigma \rightarrow_{\mathcal{F}_B} \Sigma$ of n elements is defined as

$$x_p = \min_{i<n} \{\sigma(i) : |\{j < n : \sigma(j) < \sigma(i)\}| \geq (1-p)n\}.$$

That is, the minimum element of σ such that at least a $(1-p)$ fraction of elements in σ are below it. Since the CDF of the input distribution $\mathcal{F}_B$ is unknown in advance, we need to obtain an empirical estimate $\hat{x}_p$ of the p-quantile. We shall do so using the first m readings of σ. For differential privacy, the estimate $\hat{x}_p$ needs to be *stable*.[6] From Proposition 2, this means that we require $m = \Omega(\frac{1}{p})$ readings. On the other hand, for a continual release application, m should be small compared to n. More specifically, the time lag m should satisfy

$$n \gg m \gg \frac{1}{p}. \tag{4}$$

Additionally, to minimise outlier error, i.e., to minimise β_{out}, p needs to be small.

A. Informal Roadmap

Since the empirical p-quantile, i.e., $\hat{x}_p$, reaches x_p in expectation, it is not possible to upper bound the probability $\Pr[\hat{x}_p < x_p]$ arbitrarily to minimise errors due to truncation (step 5 of Mechanism 1). We therefore introduce another parameter $\lambda < 1$, and seek to estimate the λp-quantile (instead). This allows us to bound $\Pr[\hat{x}_{\lambda p} < x_p]$ by adjusting the parameter λ, since $x_{\lambda p} > x_p$ if $\lambda < 1$. We denote this probability bound by β_{qt}, shown in Figure 4. To make the estimate differentially private, and consequently to use it as the threshold τ, we may use additive Laplace noise. Due to the properties of the Laplace distribution, $\Pr[\tau < x_p]$ is non-zero. We seek to bound this within β_{lt}. Finally, to fix τ below a bound $\tau_{\max}$, we set the upper bound β_{rt} on the probability that $t > \tau_{\max}$. The ability to bound these three error probabilities is important for our utility analysis. In the following, we will formally introduce these sources of errors and will subsequently try to minimise them for utility.

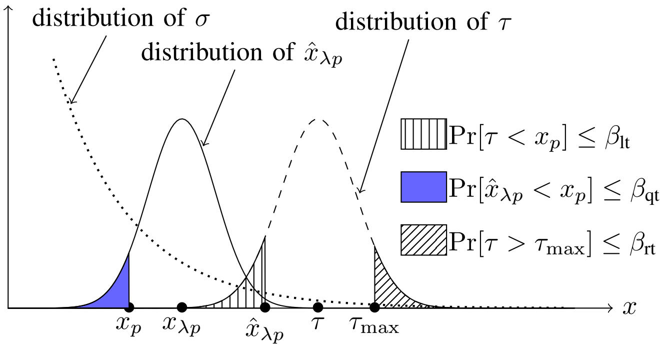

distribution of σ
distribution of $\hat{x}_{\lambda p}$
distribution of τ
▥ $\Pr[\tau < x_p] \leq \beta_{\text{lt}}$
■ $\Pr[\hat{x}_{\lambda p} < x_p] \leq \beta_{\text{qt}}$
▨ $\Pr[\tau > \tau_{\max}] \leq \beta_{\text{rt}}$
x
x_p
$x_{\lambda p}$
$\hat{x}_{\lambda p}$
τ
$\tau_{\max}$

Fig. 4. Possible sources of error when estimating the threshold τ using the p-quantile.

B. Error due to Underestimating the λp-Quantile

As mentioned above, since the expected value of $\hat{x}_p$ is x_p, we cannot bound $\Pr[\hat{x}_p < x_p]$ below and arbitrary bound β_{qt}. Thus, instead of estimating the p-quantile, we shall estimate the λp-quantile with $\frac{1}{pm} < \lambda < 1$. Now the probability of having $\hat{x}_{\lambda p} < x_p$ is given by:

$$\begin{aligned} g(\lambda, p, m) &= \Pr[\hat{x}_{\lambda p} < x_p] \\ &= \Pr[< \lambda pm \text{ values of } \sigma \text{ are } \geq x_p] \\ &= \sum_{i=0}^{\lambda pm} \binom{m}{i} p^i (1-p)^{m-i}. \end{aligned} \tag{5}$$

We denote by β_{qt} the bound on the error probability function g.

C. Privately Obtaining the λp-Quantile

For this section we assume that $x_p \leq \hat{x}_{\lambda p}$ holds. As discussed before, setting the threshold τ to $\hat{x}_{\lambda p}$ is not private. To obtain a differentially private estimate, we utilize smooth sensitivity. As shown in [10], smooth sensitivity can be used to display the median in a differentially private manner. We modify the median algorithm described therein to privately release the λp-quantile. First we compute the smooth sensitivity of the empirical λp-quantile, i.e., $\hat{x}_{\lambda p}$, as

$$\mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}) = \max_{k=0,1,\ldots,m+1} \{e^{-bk} \mathsf{LS}_{\sigma'}(\hat{x}_{\lambda p}) : d(\sigma, \sigma') \leq k\},$$

where

$$\mathsf{LS}_{\sigma'}(\hat{x}_{\lambda p}) = \max_{t=0,1,\ldots,k+1} |\ddot{\sigma}(P+t) - \ddot{\sigma}(P+t-k-1)|.$$

Here, $\ddot{\sigma}$ is the sorted string of the first m values of σ in ascending order with 0 added as a prepend and B as an appendix; and P is the rank of $\hat{x}_{\lambda p}$. This can be done in $O(m^2)$ time [10, §3.1].

Warm: After computing the smooth sensitivity, we can set the threshold τ as

$$\tau = \hat{x}_{\lambda p} + \frac{\mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})}{a} \cdot \text{noise},$$

where noise is either the Laplace or standard Gaussian noise. For both, we can set $b \leq \frac{\epsilon}{-2\log(\delta)}$ as the smoothing parameter (Definition 10. If we use the Laplace distribution with scale 1, $a = \frac{\epsilon}{2}$ results in (ϵ, δ)-differential privacy. When the noise is standard Gaussian, then $a = \frac{\epsilon}{\sqrt{-\ln\delta}}$ gives us (ϵ, δ)-differential privacy [10]. However, as discussed in Section IV-A, we require $\Pr[\tau < x_p]$ to be bounded by an arbitrary β_{lt} (see Figure 4). Unfortunately, the only way to bound this probability is by adjusting the privacy parameter ϵ, which cannot be arbitrarily chosen (without compromising privacy). Thus, we need to slightly change the above estimate.

Warmer: Let G_{ns} denote the CDF of the noise distribution. Then instead of the above we can set τ to

$$\begin{aligned} \tau &= \hat{x}_{\lambda p} + \frac{\mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})}{a} \cdot (\text{noise} + \text{offset}) \\ &= \hat{x}_{\lambda p} + \frac{\mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})}{a} \cdot \text{offset} + \frac{\mathsf{SS}_{\sigma,b}(\hat{x}_{p})}{a} \cdot \text{noise}. \end{aligned}$$

where offset $= G_{\text{ns}}^{-1}(1 - \beta_{\text{lt}})$. That is, we offset the noise to the right of x_p to ensure that the probability of the threshold

[6] A real-valued function f of σ is said to be k-stable if adding or removing any k elements from σ does not change the value of f. See [3, §7].

τ falling below x_p is bounded by β_{lt}. Unfortunately, this is no longer differentially private, since the offset itself may leak information.

Solution: Informally, to solve this problem we will multiplicatively increase $\mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})$ by a factor κ such that it "hides" the offset as well.[7] First define τ' to be

$$\tau' = \hat{x}_{\lambda p} + \frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})}{a} \cdot G_{\text{ns}}^{-1}(1-\beta_{\text{lt}}), \tag{6}$$

where κ is a positive real number to be determined. Then, we finally set

$$\tau = \tau' + \frac{\mathsf{SS}_{\sigma,b}(\tau')}{a} \cdot \text{noise}. \tag{7}$$

We seek the smallest κ such that the probability of having $\tau < x_p$ is bounded by β_{lt}. Now, to bound this probability a little algebraic manipulation using Eqs. 6 and 7, together with our assumption $\hat{x}_{\lambda p} \geq x_p$ (beginning of this section), shows that having

$$\Pr\left[\text{noise} < -\frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}) G_{\text{ns}}^{-1}(1-\beta_{\text{lt}})}{\mathsf{SS}_{\sigma,b}(\tau')}\right] \leq \beta_{\text{lt}},$$

suffices. This is equivalent to

$$\begin{aligned} G_{\text{ns}}\left(-\frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}) G_{\text{ns}}^{-1}(1-\beta_{\text{lt}})}{\mathsf{SS}_{\sigma,b}(\tau')}\right) &\leq \beta_{\text{lt}} \\ \Leftrightarrow -\frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}) G_{\text{ns}}^{-1}(1-\beta_{\text{lt}})}{\mathsf{SS}_{\sigma,b}(\tau')} &\leq G_{\text{ns}}^{-1}(\beta_{\text{lt}}) \\ \Leftrightarrow \frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}) G_{\text{ns}}^{-1}(1-\beta_{\text{lt}})}{\mathsf{SS}_{\sigma,b}(\tau')} &\geq G_{\text{ns}}^{-1}(1-\beta_{\text{lt}}) \\ \Leftrightarrow \kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}) &\geq \mathsf{SS}_{\sigma,b}(\tau'). \end{aligned} \tag{8}$$

Also, from Eq. 6, using the triangle inequality and homogeneity property of smooth sensitivity [10], we get

$$\mathsf{SS}_{\sigma,b}(\tau') \leq \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}) + \frac{\kappa G_{\text{ns}}^{-1}(1-\beta_{\text{lt}})}{a} \cdot \mathsf{SS}_{\sigma,b}(\mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})) \tag{9}$$

From the definition of smooth sensitivity we see that $\forall \sigma, \sigma'$ such that $d(\sigma, \sigma') = 1$, we have

$$\mathsf{SS}_{\sigma',b}(c) \leq e^b \mathsf{SS}_{\sigma,b}(c).$$

This also allows us to write, $\forall \sigma'$ such that $d(\sigma, \sigma') = 1$

$$\mathsf{LS}_{\sigma'}(\mathsf{SS}_{\sigma,b}(c)) \leq (e^b - 1)\mathsf{SS}_{\sigma',b}(c).$$

Now similar to the computation of smooth sensitivity of the median in [10], we have

$$\begin{aligned} &\mathsf{SS}_{\sigma,b}(\mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})) \\ &\leq \max_{k \in \mathbb{N}} \{e^{-bk} \mathsf{LS}_{\sigma'}(\mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})) : d(\sigma,\sigma') \leq k\} \\ &\leq \max_{k \in \mathbb{N}} \{e^{-bk} (e^b - 1) \mathsf{SS}_{\sigma',b}(\hat{x}_{\lambda p}) : d(\sigma,\sigma') \leq k\} \\ &\leq \max_{k \in \mathbb{N}} \{e^{-bk} (e^b - 1) e^{bk} \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}) : d(\sigma,\sigma') \leq k\} \\ &\leq \max_{k \in \mathbb{N}} \{(e^b - 1) \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}) : d(\sigma,\sigma') \leq k\} \\ &\leq (e^b - 1) \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}). \end{aligned} \tag{10}$$

Equating Eqs. 8, 9 and 10 gives us the required κ as

$$\kappa = \left(1 - \frac{(e^b - 1) G_{\text{ns}}^{-1}(1-\beta_{\text{lt}})}{a}\right)^{-1}. \tag{11}$$

Now, putting Eq. 7 into Eq. 6, and using Eq. 8 we have

$$\begin{aligned} \tau &= \hat{x}_{\lambda p} + \frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})}{a} \cdot G_{\text{ns}}^{-1}(1-\beta_2) + \frac{\mathsf{SS}_{\sigma,b}(\tau')}{a} \cdot \text{noise} \\ &\leq \hat{x}_{\lambda p} + \frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})}{a} \cdot G_{\text{ns}}^{-1}(1-\beta_2) \\ &\quad + \frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})}{a} \cdot \text{noise} \\ &= \hat{x}_{\lambda p} + \frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})}{a} \cdot (\text{noise} + G_{\text{ns}}^{-1}(1-\beta_{\text{lt}})), \end{aligned} \tag{12}$$

where κ is given by Eq. 11. The threshold τ released via the above mechanism is differentially private since $\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})$ is a smooth upper bound of $\hat{x}_{\lambda p}$ and κ only depends on public parameters.

D. Upper Bound on the Threshold

For our utility analysis in the next section, we require an upper bound on the random variable τ from Eq. 12. We see that with probability at least $1 - \beta_{\text{rt}}$, we have

$$\tau \leq \tau_{\max} = \hat{x}_{\lambda p} + \frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})}{a} \cdot (G_{\text{ns}}^{-1}(1-\beta_{\text{lt}}) + G_{\text{ns}}^{-1}(1-\beta_{\text{rt}})) \tag{13}$$

V. Utility Analysis

For this section, we assume that the λp-quantile has been obtained after m steps and satisfies the constraint $x_p \leq \hat{x}_{\lambda p}$. Furthermore, a threshold τ has been obtained via Eq. 12 satisfying Eq. 13. As described in Section III-C, our mechanism (Mechanism 1) then releases c on every new observation from $m+1$ to n. For the ith observation $x = \sigma(i)$ where $i > m$, if $x \leq \tau$ then we release $c(\sigma, i)$ through the BT algorithm with noise $\text{Lap}(\frac{\tau \log_2(n-m)}{\epsilon})$. This causes an additive error α_{Lap} in the computation of c with an associated error probability β_{Lap}. On the other hand, if $x > \tau$, we instead assume that the new observation is exactly τ and then again add noise as before. This induces an additional error term, which we have called outlier error, denoted α_{out}. We denote the probability of the outlier error by β_{out}. In the following, we bound these two errors by first assuming the (unrealistic) worst case scenario, i.e., every new observation after the time lag m steps is exactly B with probability p. We then use the more realistic assumption that the distribution of the stream is light-tailed, and show that based on real-world datasets we are expected to gain significant utility in practice.

A. Worst Case Error

Let ξ denote the PDF of the outlier error and Ξ its CDF. Let E be a random variable denoting the outlier error and let $E_i = \sigma(i) - \min\{\sigma(i), \tau\}$ denote the outlier error of observation i, which is bounded by $B - \tau$. Assuming each element of σ is distributed as $X \sim \mathcal{F}_B$ (cf. Definition 4), we have $\Xi(x) \geq F(x - x_p)$ for strictly positive $x \in X$. The worst case is when the PDF is given as

$$\xi(x) = \Delta(x)(1-p) + \Delta(x - (B - \tau))p,$$

[7] Thus aiming for a smooth upper bound on smooth sensitivity, rather than smooth sensitivity itself.

where Δ is the Dirac delta function. This means that beyond the p-quantile, all the values are equal to B. With this assumption we can estimate the $(\alpha_{\text{out}}, \beta_{\text{out}})$-utility (Definition 5) as follows:

$$\begin{aligned}\Pr\left[\sum_{i=1}^{n} E_i \geq \alpha_{\text{out}}\right] &= \Pr\left[h\sum_{i=1}^{n} E_i \geq h\alpha_{\text{out}}\right] \\ &= \Pr\left[\exp\left(h\sum_{i=1}^{n} E_i\right) \geq \exp(h\alpha_{\text{out}})\right] \\ &\leq \frac{\mathbb{E}[\exp(h\sum_{i=1}^{n} E_i)]}{\exp(h\alpha_{\text{out}})} \\ &= \frac{\prod_{i=1}^{n} \mathbb{E}[\exp(hE_i)]}{\exp(h\alpha_{\text{out}})} \\ &= \frac{\mathbb{E}[\exp(hE)]^n}{\exp(h\alpha_{\text{out}})} \\ &= \frac{(1-p+pe^{h(B-\tau)})^n}{e^{h\alpha_{\text{out}}}} = \beta_{\text{out}}.\end{aligned}$$

Solving for α_{out}, we get

$$\alpha_{\text{out}} = \frac{n\ln(1-p+pe^{h(B-\tau)}) + \ln\frac{1}{\beta_{\text{out}}}}{h}.$$

The value of $h \approx \frac{1}{(B-\tau)pn}$ minimizes α_{out}. Recall that according to Eq. 4, we want $m \gg \frac{1}{p}$, which implies that $h \ll 1$. The above then becomes

$$\alpha_{\text{out}} = pn(B-\tau)\left(\ln\frac{1}{\beta_{\text{out}}} + 1\right) + o(1).$$

Adding this to the utility term α_{Lap} from the BT algorithm (Eq. 3) [5] we see that the overall error α is

$$\begin{aligned}\alpha \leq\ & \frac{1}{\epsilon}(\log_2(n-m))^{1.5}\tau\sqrt{8\ln\frac{1}{\beta_{\text{Lap}}}} \\ &+ (B-\tau)pn\left(\ln\frac{1}{\beta_{\text{out}}} + 1\right), \qquad (14)\end{aligned}$$

with probability at most β, where β is a bound on the sum of the five error probabilities.[8] If the error is dominated by the first summand, then this leads to an improvement factor of $\frac{B}{\tau}$ in utility over the application of the BT algorithm without our mechanism (see Eq. 3). However, looking at the second summand, we see that the outlier error is proportional to pn. To make this into a constant error term, we need $p \approx \frac{1}{n}$. But recall from Eq. 4 that we require $m \gg \frac{1}{p}$. Thus, it is not possible to bound this error term. Hence, if the input stream has the worst-case distribution, our mechanism does not improve utility. However, arguably, real-world data streams are not distributed in this way.

B. Error on Light-Tailed Distributions

As shown in Section III-B, many real-world data distributions are expected to be light-tailed (cf. Definition 12), thus behaving significantly differently than the worst case. More precisely, we focus on distributions that are light-tailed beyond their $p_{\max}$-quantile, a quantity to be determined shortly. Clearly, this holds true for any $p \leq p_{\max}$ as well. Figure 5 shows that this assumption holds for the train trips dataset for $p = 0.005$-quantile. The figure shows the ECDF of travel times against the CDF of the exponential distribution with parameter $\gamma = \frac{-\ln p}{x_p} = \frac{-\ln 0.005}{x_{0.005}}$ (cf. Fact 1). The assumption also holds for the supermarket dataset with the same p-quantile. We omit the graph due to repetition.

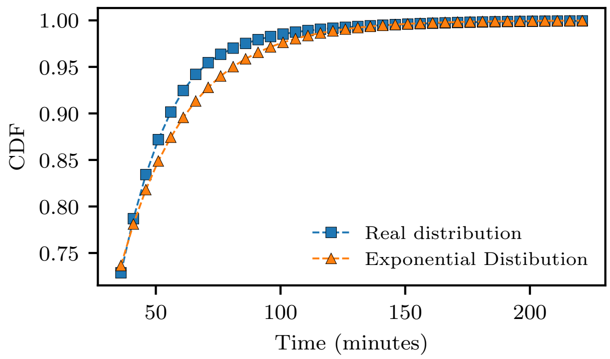


Fig. 5. The distribution of the train trips dataset compared to an exponential distribution with the same $p = 0.005$-quantile. The real distribution is above the exponential distribution, indicating that it is light-tailed.

Since the distribution is light-tailed, we can use Proposition 1 and the assumption $x_p \leq \hat{x}_{\lambda p}$ to conclude that for all $p \leq p_{\max}$,

$$\hat{x}_{\lambda p} \cdot r \geq x_{p^r}, \forall r \geq 1. \qquad (15)$$

Thus, instead of using the threshold τ directly from Eq. 7, we multiply it by r and set it as the threshold. According to the above equation, this results in reduced outlier error whenever $r > 1$.

Determining $p_{\max}$*:* To determine the value of $p_{\max}$, we perform a series of experiments on the train trips and the supermarket dataset. We seek a value of $p_{\max}$ that ensures the light-tailed property on both datasets. We vary p^r beginning from a value of 0.1 to increasingly small values. Against each value of p^r we obtain the empirical p^r-quantile, i.e., $\hat{x}_{p^r}$. We then use different values of p, e.g., 0.1, 0.01, and so on. From each pair of values of p and p^r, we obtain r and multiply it with the empirical p^r-quantile to obtain $r\hat{x}_p$. The aim is to find a value of $p_{\max}$ such that for all $p \leq p_{\max}$, $r\hat{x}_p \approx \hat{x}_{p^r}$. Since $\hat{x}_{\lambda p} \geq \hat{x}_p$, this implies that Eq. 15 would be satisfied. The results are shown in Figure 6a for the train trips dataset and Figure 6b for the supermarket dataset. The results suggest that $p_{\max} \approx 0.005$ suffices.

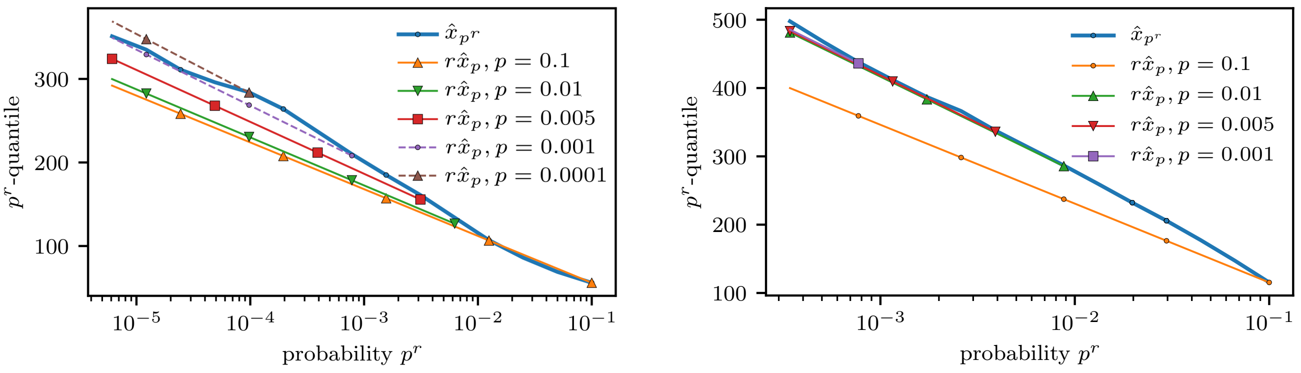
Error Bound: Now, assuming that the input stream is light-tailed we see that the outlier error is bounded by the properties of the exponential distribution. That is, PDF ξ of the outlier error can be written as

$$\xi(x) = \Delta(x)(1-p^r) + p^r \cdot \gamma e^{-\gamma x},$$

where $\gamma = \frac{-\ln p}{x_p}$. Thus,

$$\Pr\left[\sum_{i=1}^{n} E_i \geq \alpha_{\text{out}}\right] \leq \frac{\mathbb{E}[\exp(hE)]^n}{\exp(h\alpha_{\text{out}})} \leq \frac{(1+\frac{h}{\gamma-h})^n}{e^{h\alpha_{\text{out}}}} = \beta_{\text{out}}.$$

[8] i.e., β_{qt}, β_{lt}, β_{Lap}, β_{out} and β_{rt}.



(a) Train trips dataset

(b) Supermarket dataset

Fig. 6. Error in estimating the empirical p^r-quantile through empirical p-quantile with different choices of p. We see that below $p_{\max} \approx 0.005$, the input datasets satisfy $\hat{x}_p \cdot r \approx x_{p^r}$.

This gives us,

$$\alpha_{\text{out}} = \frac{n \ln(1 + \frac{h}{\gamma - h}) + \ln(\frac{1}{\beta_{\text{out}}})}{h}$$

We can choose the h that minimises α_{out} as

$$h = \gamma \left(\sqrt{\frac{p^r \cdot n}{\ln \frac{1}{\beta_{\text{out}}}}} + 1 \right)^{-1},$$

which leads to

$$\alpha_{\text{out}} \leq \frac{-x_p}{\ln p} \left(\sqrt{p^r \cdot n} + \sqrt{\ln \frac{1}{\beta_{\text{out}}}} \right)^2 .$$

Adding this to the error term α_{Lap} from the BT algorithm (Eq. 3) and using the assumption $x_p \leq \hat{x}_{\lambda p}$, we see that the overall error α is

$$\alpha \leq \frac{1}{\epsilon} (\log_2(n-m))^{1.5} \tau r \sqrt{8 \ln \frac{1}{\beta_{\text{Lap}}}} + \frac{-\hat{x}_{\lambda p}}{\ln p} \left(\sqrt{p^r \cdot n} + \sqrt{\ln \frac{1}{\beta_{\text{out}}}} \right)^2, \tag{16}$$

with probability at least $1 - \beta$, where β is once again a bound on the five error probabilities. Now, to bound the second error term (the second summand) by a constant, we require $p^r \approx \frac{1}{n}$. Thus an r logarithmic in n suffices. With this value of r we see that the overall error is bounded by $O(\tau(\log_2 n)^{1.5}/\epsilon)$. Thus, we obtain an improvement factor of B/τ over the BT algorithm, which was the aim of our mechanism. In the next section, we will show how to optimize the parameters for utility.

VI. Optimizing Utility

A. Optimized Parameters

To optimize α given by Eq. 16, we ran a series of experiments on the two datasets using the Python library SciPy.[9] Specifically, we used a truncated Newton method [12] (TNC) implemented by the `scipy.optimize.minimize` method to optimize α. We fix $n =$ 25,000,000 for the train trips dataset and $n =$ 150,000 for the supermarket dataset. The parameter β, i.e., overall probability of exceeding an error of α, was fixed to 0.02, and δ was fixed to 2^{-20}.[10] For both datasets, we analyze the influence of local and global parameters separately.

Effect of Local Parameters: We fixed $\epsilon = 1$ for this series of experiments. Then, for different values of the time lag m, we ran the optimizer on the objective function α given by Eq. 16, with the constraints: $p \leq p_{\max} = 0.005$, $\lambda \leq 1$, $r \geq 1$, and $\kappa > 0$. Note that the optimization algorithm is deterministic: given fixed global parameters, we obtain the same value of the local parameters each time. We also define the improvement factor (IF), as the ratio of error obtained from the BT algorithm to the error obtained through our mechanism. The results are shown in Tables I and II.

TABLE I. Optimized parameters for the train trips dataset

m	IF	r	λ	p	$\frac{\epsilon_1}{\epsilon}$	$\frac{\beta_{\text{qt}}}{\beta}$	$\frac{\beta_{\text{lt}}}{\beta}$	$\frac{\beta_{\text{Lap}}}{\beta}$	$\frac{\beta_{\text{out}}}{\beta}$	$\frac{\beta_{\text{rt}}}{\beta}$
40000	1	1	1	0.005	0	0.00	0.00	1	0.00	0.00
50000	2.32	1.63	0.85	0.005	0.8	0.09	0.3	0.41	0.09	0.09
60000	2.8	1.49	0.83	0.0049	0.9	0.13	0.23	0.37	0.13	0.13
100000	3.69	1.76	0.87	0.0048	0.9	0.07	0.1	0.68	0.07	0.07
300000	4.34	1.91	0.87	0.005	0.79	0.11	0.11	0.57	0.11	0.11

TABLE II. Optimized parameters for the supermarket dataset

m	IF	r	λ	p	$\frac{\epsilon_1}{\epsilon}$	$\frac{\beta_{\text{qt}}}{\beta}$	$\frac{\beta_{\text{lt}}}{\beta}$	$\frac{\beta_{\text{Lap}}}{\beta}$	$\frac{\beta_{\text{out}}}{\beta}$	$\frac{\beta_{\text{rt}}}{\beta}$
40000	1	1	1	0.005	0	0.00	0.00	1	0.00	0.00
50000	4.23	1	0.81	0.005	0.82	0.23	0.19	0.19	0.19	0.19
60000	4.86	1.08	0.82	0.0049	0.88	0.15	0.27	0.27	0.15	0.15
100000	6.67	1.11	0.85	0.0046	0.81	0.07	0.07	0.71	0.07	0.07

We see that as the time lag m grows, IF is less impacted by Laplace noise added to the sum as indicated by the decreasing ratio β_{Lap}/β. The improvement factor grows with increasing m, but this is essentially a trade-off between releasing or withholding the sum. The rest of the parameters are relatively stable, with higher values of r indicating that the threshold can be set higher than the estimated experimental λp-quantile. For smaller m, we do not see any improvement in utility over the basic BT algorithm. Note that in such a case our mechanism

[9] https://www.scipy.org/

[10] While this value of δ is higher than recommended (i.e., negligible in $1/n$ [3, §2.3, p. 18]), lower values, say 2^{-30} [13, §3, p. 5], have a minor impact on utility in our experiments.

simply releases the sum using the BT algorithm with noise scaled to B. Thus, we do not incur any extra cost in utility.

Effect of Global Parameters: The parameters ϵ, δ, β, and m are global parameters specified to the optimization algorithm. The parameters β and δ are fixed as before. Thus, we look at the evolution of IF with different values of ϵ and m. For each value, we run the optimizer to output a set of local parameters that maximize utility. For this, we only use the train trips dataset with $n = 25{,}000{,}000$.

- ϵ: Smooth sensitivity is roughly proportional to $\frac{1}{\epsilon_1^2}$, so smaller values of ϵ (and consequently ϵ_1) will not result in a high IF. On the other hand, if ϵ is too large, the error caused by truncation (step 5 of Mechanism 1) will overwhelm the noise due to the Laplace mechanism, and hence the IF will be low. Figure 7 (left) shows this trend, where we plot the improvement factor of our algorithm over the BT algorithm by fixing $m = 50{,}000$.
- m: The impact of the time lag m is data dependent. We do not see much improvement when m is small, say around 10,000. With m around 50,000 we see noticeable increase in IF. This is indicated by Figure 7 (right), where we have fixed $\epsilon = 1$.

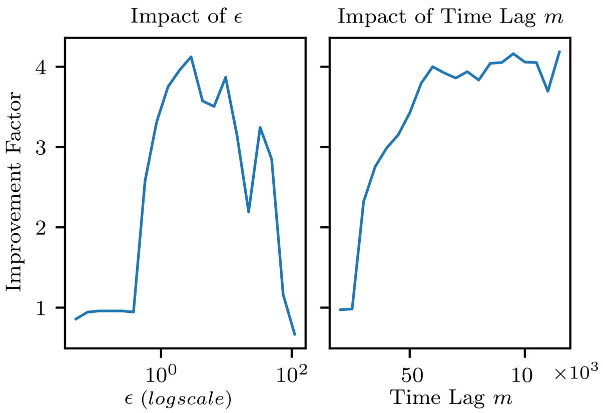


Fig. 7. Influence of global parameters on the improvement factor.

B. Private Choice of Parameters

The parameters required as input in Mechanism 1 are ϵ_1, ϵ_2, δ, m and r. For reasons of privacy, the choice of these parameters cannot be based on optimization on a particular input dataset. We therefore discuss some heuristic choices for these parameters based on our experiments above. The parameters $\epsilon = \epsilon_1 + \epsilon_2$ and δ can be chosen in the standard way. For instance, $\epsilon = 1$ and $\delta = n^{-2}$. From Tables I and II, a value in the range 0.8 to 0.9 is a reasonable choice for ϵ_1. Note that ϵ_2 is readily determined by ϵ and ϵ_1. From the same tables, we see that an r between 1 and 2 suffices. We therefore discuss heuristics for choosing m.

Heuristics for Selecting the Time Lag m: We specify two criteria that should be satisfied by the time lag. By assigning reasonably conservative (with respect to utility guarantees) values to the free parameters in the two criteria, we obtain a value of m that is expected to provide good utility in practice. The overall value of m can be obtained as the maximum of the two values returned by the criteria.

a) First Criterion: This is the probability of having $\hat{x}_{\lambda p} < x_p$ given by the function g in Eq. 5. We set $g(\lambda, p, m) = g(0.5, p_{\max}, m) < \beta$. All our optimization experiments returned a value of λ close to 1. Thus, setting $\lambda = 0.5$ is a reasonably conservative choice.

b) Second Criterion: This is related to the (approximate) scale of smooth sensitivity:[11]

$$\frac{\kappa \mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})}{a} \cdot G_{\text{ns}}^{-1}(1-\beta) \approx \frac{B}{10}. \tag{17}$$

The term $B/10$ is arbitrarily chosen to ensure that the scale is a few orders of magnitude less than B. With the Laplace noise distribution, we can take $a = \frac{\epsilon}{\sqrt{-\ln \delta}}$ and $b = \min(1, \frac{\epsilon}{-2\ln\delta})$. This readily gives us κ through Eq. 11. Now, to get a conservative bound on $\mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p})$, we first assume that the exponential distribution has its $p_{\max}$-quantile close to B. In other words, $x_{p_{\max}} \approx B$. This means that the upper bound on the smooth sensitivity is conservative. Now at B, the probability density function of the exponential distribution is $\frac{-p_{\max}\ln p_{\max}}{B}$. With m observations, the inverse of the average distance between two observations is therefore roughly

$$d = \frac{-mp_{\max}\ln p_{\max}}{B}. \tag{18}$$

Now, assuming that the density is approximately constant in the neighbourhood of B, smooth sensitivity is given by

$$\max_{k\in\mathbb{N}}\{dk\exp(-bk)\}.$$

We can bound the above if we replace natural numbers with real numbers, resulting in

$$\mathsf{SS}_{\sigma,b}(\hat{x}_{\lambda p}) < \frac{d\exp(-1)}{b}. \tag{19}$$

Now, combining Eqs. 17, 18 and 19 and solving for m, we get our second criterion on m as

$$m > \frac{20\kappa(-\ln\delta)^{1.5}\exp(-1)G_{\text{ns}}^{-1}(1-\beta)}{-\epsilon^2 p_{\max}\ln p_{\max}}.$$

With $\beta = 0.02$, $\delta = 2^{-20}$, $\epsilon = 1$ and $p_{\max} = 0.005$, we find that m should be at least 20,000 in order to satisfy the first criterion. The second criterion imposes $m > 80,000$ for good utility. From Tables I and II we see that with $m > 80,000$ we indeed obtain good utility. Of course, the higher the time lag m, the better the utility gain, with the trade-off that there is a longer time lag before we output the sum.

VII. Experimental Evaluation

A. Accumulated Error on the Sum

We now show the improvement factor in computing the moving average (sum) through our mechanism. Since the error is maximized at step n, i.e., the last observation of the stream, we compare the value of $\hat{c}(\sigma, n)$ through our mechanism against its counterpart via the BT algorithm. For both datasets, we run the two mechanisms a total of 20,000 times and display the empirical probability density function (PDF) of error.

[11] If the time lag is small, the observations will be sparsely distributed implying a large scale of smooth sensitivity.

1) Train Trips Dataset: We fixed $\epsilon = 1$, $\delta = 2^{-20}$, $\beta = 0.02$, $m = 50,000$ and $n = 25,000,000$, and obtained the values of the local parameters after optimization, shown in Table I. We set $B = 1440$ minutes, which is the maximum possible commute time in a 24 hour period. Figure 8 shows the PDF of the resulting error (normalized by the maximum value B) of our mechanism and the BT algorithm. We see that the error in our case is more tightly concentrated around 0. On average, we obtain an improvement factor of 3.5.

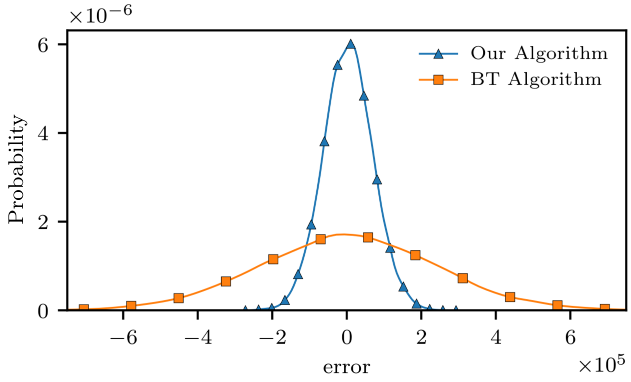


Fig. 8. PDF of the error on the train trips dataset through our mechanism and the BT algorithm.

We are also interested in knowing whether our estimation of the outlier error (i.e., the second summand in Eq. 16), is close to the actual outlier error. This will validate whether our assumption that the distribution of the dataset is light-tailed. To verify this we re-ran our mechanism 20,000 times on the same dataset and obtained the ratio $\frac{\text{real error}}{\text{estimated error}}$ with the same parameters as above. Figure 9 shows that we have erred on the precautionary side with our estimation of outlier error being well within the actual error.

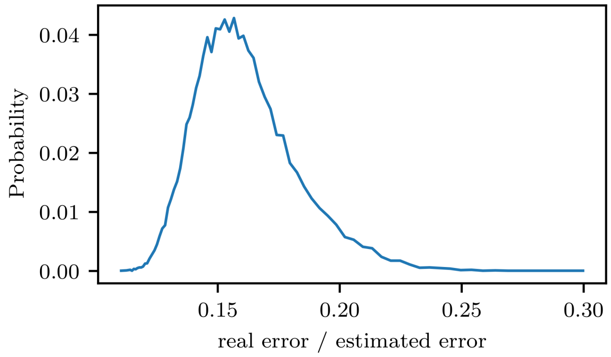


Fig. 9. Distribution of the outlier error ratio: (real error)/(estimated error) on the train trips dataset. Our estimated outlier error is well below the real outlier error.

2) Supermarket Dataset: For the supermarket dataset, we use the same set of parameters except that we have $n = 150,000$ (due to less data points) and $B = 3,000$ dollars (a conservative guess on the amount spent). Figure 10 shows the PDF of the error from our mechanism and the BT algorithm. Once again the error through our mechanism is more tightly concentrated around 0. For this dataset, we perform much better than the BT algorithm, with an improvement factor of 9 on average.

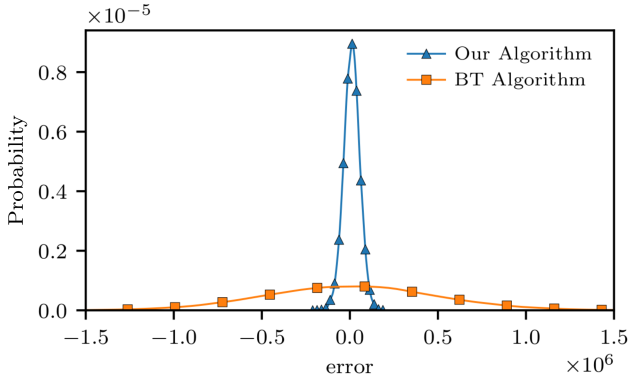


Fig. 10. PDF of the error on the supermarket dataset through our mechanism and the BT algorithm.

For this dataset as well we are interested in knowing whether the estimated error due to outliers is well below the actual error. Figure 11 shows that the dataset does indeed have a light-tailed distribution with the actual error being almost always below the estimated value.

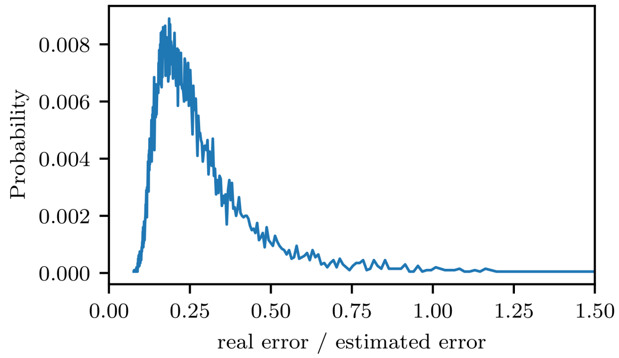


Fig. 11. Distribution of the outlier error ratio: (real error)/(estimated error) on the supermarket dataset. Once again our estimated outlier error is close to the real outlier error.

B. Does the Distribution Remain Light-Tailed across Time?

Recall that our mechanism promises improved utility based on the premise that data distribution is light-tailed. Since the input stream is time dependent, the estimated threshold (using the λp-quantile) through m observations with a given time period may be drastically different from its estimate via a different time period. To ensure that this is not the case, we analyzed the distribution of the train trips dataset across different hours and different days of the week. The distributions are shown in Figures 12 and 13, respectively. While the beginning of the distributions show variation based on the time period, the tails are similar and light-tailed. Thus, our estimated threshold is likely to improve utility independent of the time period in real datasets.

VIII. Related Work

As previously noted, the privacy-preserving algorithms for continual release of statistics from binary streams proposed

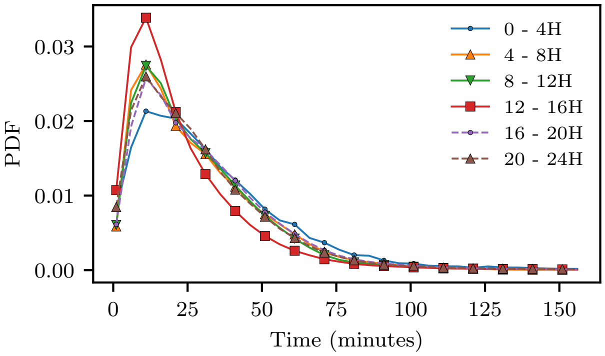


Fig. 12. Train commute time distribution from different hours of the day. Again all are light-tailed.

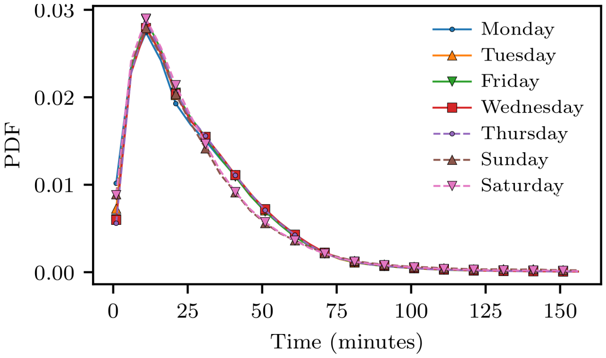


Fig. 13. Train commute time distribution from different days of the week. All are light-tailed.

in [4], [5] can be generalized to the scenario addressed in this paper, i.e., release of statistics from a stream whose values are from the real interval $[0, B]$. Indeed, we have used the algorithm from [5] as one of the components of our method. However, the focus of the two works in [4] and [5] is on improving the error for binary strings which do not have the added factor of B. The two algorithms are based on event-level privacy. As such, if the aim of privacy is to protect all events from an individual (e.g., all trips made by an individual over the course of the whole year), then the privacy provided by these algorithms is insufficient. The work from [14] attempts to improve this by offering privacy for up to w successive events. Noting that any w successive events might not contain multiple events originating from a single individual, the authors from [15] introduce l-trajectory privacy, where any successive l events from a user are targeted for privacy. These works essentially propose privacy mechanisms for variants of the definition of differential privacy where neighbouring streams are defined differently from the standard definition of Hamming distance. We note that our method can be easily used in conjunction with these algorithms, as we only use the BT algorithm from [5] in a modular way. However, to find a utility maximizing threshold in a differentially private manner for any variation in the definition of neighbouring streams requires tweaking our mechanism. Likewise, these algorithms also target infinite streams as opposed to bounded streams (as is done in our paper). Application of our approach to these settings is an interesting area for future work.

As argued before, privacy-preserving continual release of the sum is only one example of functions that can be released with improved utility through our mechanism. As long as the target function remains a function of the stream, has reduced sensitivity based on tighter concentration of input data, and the error due to outliers can be bounded and related to the p-quantile, we can adapt all the steps of our method to the given function. This allows us to estimate the threshold from the data, thus finding the optimum balance between the error due to symmetric noise and the error due to outliers. Examples of such functions include the sliding window average or the decaying sum where either past observations are completely discarded or are given progressively less weights [7]. Another example is continually releasing histogram of the input stream where we would like to completely discard bins above the main concentration of the data.

Our work on estimating the threshold using the p-quantile can be thought of as an exercise in finding "robust" statistics [16] with differential privacy, a line of work that was discussed in [17] and [18]. These works estimate the scale of the input data with the help of the interquartile range using the Propose-Test-Release approach [17]. Briefly, this approach checks whether a given analysis uses a function that is robust or stable [9] on a given dataset or not. If the answer is no, the analysis is abandoned. In other words, the interquartile range may not even be released if it is not stable for the given dataset. Our approach is different as we use the p-quantile as the estimate of the scale of the input dataset, and use smooth sensitivity to release it. Unlike the Propose-Test-Release approach which may not release the p-quantile depending on input data, we have the advantage that we always obtain an estimate. This allows us to optimize utility by bounding errors introduced by the estimation of the scale of the dataset. We note that the problem of finding differentially private quantiles is also tackled in [19], but the main ingredient there is the exponential mechanism [20], and the context is static datasets rather than continual release of data.

The main idea of our work is to reduce the sensitivity of the query (in our case, the moving sum), by relying on some initial knowledge of the input data. If we succeed in reducing sensitivity, we significantly reduce the scale of noise added to the query, thus improving accuracy of the query answer. A similar approach has also been used in some other works for other query types or applications. In [21], the authors use prior knowledge of the dataset to release time-series data with better accuracy. The aforementioned approach is also used in [8] where the aim is to display a differentially private histogram by altering the size of the bins in order to artificially reduce sensitivity.

IX. Discussion

An interesting question to ponder is what kind of data distributions are likely to have a light-tailed distribution. Looking at the two datasets evaluated in this paper, we see that one common characteristic is that they emerge from short-lived, time-constrained events. Thus, more generally, streaming data with short-lived events is likely to exhibit light-tailed distributions. In addition to the two datasets used in this paper, other examples of data exhibiting a light-tailed distribution include smart meter-based energy readings data (e.g. electricity usage), phone call durations data, length of posts/comments on online social networks (e.g., on the website Reddit), average time spent on a given location, or daily average inter-arrivals of check-in times (location-based networks). Consequently, the resulting readings are bounded, even though the bound B might be unknown in advance or only loosely known. These are in contrast to heavy-tailed distributions where (underlying) events are not short-lived or time-constrained, e.g., income distribution, file sizes in computer systems, and network traffic over a long period of time.

We would like to stress that in case the input distribution is not light-tailed, our estimated threshold τ would be closer to the global bound B. Thus, in the worst case we would be able to provide utility similar to the BT algorithm (with the disadvantage that we add a time lag m). An example of this is a uniform distribution over $[0, B]$, where the threshold τ would be close to B (estimated via the first m readings). Similar argument applies to other heavy-tailed distributions. Also, importantly, our privacy definition is not dependent on the light-tailed distribution assumption. Thus, the privacy guarantee remains the same regardless of the nature of the input distribution.

X. Conclusion

We have presented a privacy-preserving mechanism to continually display the moving average of a stream of observations where the bound on each observation is either too conservative or not known a priori. We have relied on justified assumptions on real-world datasets to obtain a better bound on observations of the stream. Moreover, we have shown how to obtain this bound in a differentially private manner while optimizing utility. Our mechanism can be applied to many real-world applications where continuous monitoring and reporting of statistics is required, e.g., smart meter data and commute times. Our techniques can be improved in several ways. We have relied on the quantile to estimate the bound on the streaming data based on smooth sensitivity. There may be other ways to display the quantile using other robust statistics. Our mechanism can be adapted to compute functions other than the moving average. Likewise, our method can be used in conjunction with algorithms that provide privacy for multiple events instead of single events as is done in this paper. Overall, we see our work as an instance of applying differential privacy in practice.

References

[1] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*. ACM, 2010, pp. 61–66.

[2] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC*, vol. 3876. Springer, 2006, pp. 265–284.

[3] 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.

[4] 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.

[5] T.-H. H. Chan, E. Shi, and D. Song, "Private and continual release of statistics," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 3, p. 26, 2011.

[6] C. Dwork and G. J. Pappas, "Privacy in information-rich intelligent infrastructure," *arXiv preprint arXiv:1706.01985*, 2017.

[7] J. Bolot, N. Fawaz, S. Muthukrishnan, A. Nikolov, and N. Taft, "Private decayed predicate sums on streams," in *Proceedings of the 16th International Conference on Database Theory*. ACM, 2013, pp. 284–295.

[8] J. Xu, Z. Zhang, X. Xiao, Y. Yang, G. Yu, and M. Winslett, "Differentially private histogram publication," *The VLDB Journal*, vol. 22, no. 6, pp. 797–822, 2013.

[9] S. Vadhan, "The complexity of differential privacy," in *Tutorials on the Foundations of Cryptography*. Springer, 2017, pp. 347–450.

[10] 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.

[11] 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.

[12] S. G. Nash, "A survey of truncated-newton methods," in *Numerical Analysis: Historical Developments in the 20th Century*. Elsevier, 2001, pp. 265–279.

[13] M. Gaboardi, J. Honaker, G. King, J. Murtagh, K. Nissim, J. Ullman, and S. Vadhan, "Psi ({\Psi}): a private data sharing interface," *arXiv preprint arXiv:1609.04340*, 2016.

[14] 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.

[15] Y. Cao and M. Yoshikawa, "Differentially private real-time data release over infinite trajectory streams," in *Mobile Data Management (MDM), 2015 16th IEEE International Conference on*, vol. 2. IEEE, 2015, pp. 68–73.

[16] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust statistics: the approach based on influence functions*. John Wiley & Sons, 2011, vol. 196.

[17] C. Dwork and J. Lei, "Differential privacy and robust statistics," in *Proceedings of the forty-first annual ACM symposium on Theory of computing*. ACM, 2009, pp. 371–380.

[18] C. Dwork and A. Smith, "Differential privacy for statistics: What we know and what we want to learn," *Journal of Privacy and Confidentiality*, vol. 1, no. 2, p. 2, 2010.

[19] A. Smith, "Privacy-preserving statistical estimation with optimal convergence rates," in *Proceedings of the forty-third annual ACM symposium on Theory of computing*. ACM, 2011, pp. 813–822.

[20] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*. IEEE, 2007, pp. 94–103.

[21] L. Fan, L. Xiong, and V. Sunderam, "Differentially private multidimensional time series release for traffic monitoring," in *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 2013, pp. 33–48.