PriStream: Privacy-Preserving Distributed Stream Monitoring of Thresholded Percentile Statistics

Jingchao Sun*, Rui Zhang†, Jinxue Zhang*, and Yanchao Zhang*
* Arizona State University, † University of Hawaii
{jcsun, jxzhang, yczhang}@asu.edu, ruizhang@hawaii.edu

***Abstract*—Distributed stream monitoring has numerous potential applications in future smart cities. Communication efficiency and data privacy are two main challenges for distributed stream monitoring services. In this paper, we propose PriStream, the first communication-efficient and privacy-preserving distributed stream monitoring system for thresholded PERCENTILE aggregates. PriStream allows the monitoring service provider to evaluate an arbitrary function over a desired percentile of distributed data reports and monitor when the output exceeds a predetermined system threshold. Detailed theoretical analysis and evaluations show that PriStream has high accuracy and communication efficiency, and differential privacy guarantees under a strong adversary model.**

I. Introduction

Distributed stream monitoring, which monitors functions over distributed and continuous data streams in real time, has great potential in future smart cities driven by the emerging Internet-of-Things paradigm. For example, with the help of a distributed mobile health monitoring system, a public health authority can monitor the health data collected by each users' mobile and wearable devices to enable various services such as public health condition monitoring, early detection of disease outbreaks, and epidemiology research studies. In addition, a waste management company can achieve cost-efficient trash collection scheduling by monitoring the sensors installed in trash containers. As another example, a utility company (e.g., electricity, natural gas, or water) can improve the efficiency and reliability of its utility infrastructure by gathering fine-grained information from the sensors at consumers' places.

Communication efficiency is a key challenge for a practical distributed stream monitoring system. In particular, the system may contain thousands of distributed sensors, e.g., in future smart-city applications. In addition, the reporting frequency should be high enough, e.g., every five minutes, to enable approximately real-time monitoring and decision making. Since most monitoring sensors are expected to have tight resource constraints, significant effort should be made to minimize their communication overhead for data reporting.

Data privacy is another major challenge for distributed stream monitoring systems. In particular, many monitoring systems rely on sensors affiliated with human users, and the raw sensor data may be sensitive in nature. For example, the data from a biomedical sensor will disclose the user's health conditions, and the data from a utility sensor can enable the profiling of the corresponding consumer's life pattern and routine. Without strong guarantee of their data privacy, users will be reluctant to join distributed monitoring systems.

There are some attempts to achieve communication-efficient and/or privacy-preserving distributed stream monitoring with aggregation thresholds. In such systems, time is divided into fixed time intervals, and each node records new data generated in each interval whereby to compute a statistic value. The goal of the monitoring service provider is to aggregate all the users' statistic values in each interval and compares the aggregation result with some predefined threshold. Such thresholded monitoring systems have important applications such as anomaly detection. Previous work sought to trade aggregation accuracy for communication efficiency by letting each node independently decide whether its data submission can contribute to the service provider's decision making; if not, the node will not submit his data. The MEAN aggregate function is addressed in [1]–[3], SUM and COUNT are considered in [1], [2], and MIN and MAX are addressed in [1]. In addition, the work in [4] incorporates differential privacy guarantees into the scheme in [3].

In this paper, we study distributed stream monitoring of thresholded PERCENTILE aggregates with high communication efficiency and also strong privacy guarantees. In particular, the monitoring service provider wants to monitor when $f(\chi_r) > \tau$ happens, where χ_r denotes the rth percentile among the statistic values from all the distributed nodes, $f(\cdot)$ denotes an arbitrary single-parameter function chosen by the service provider (say, a squaring or square root function), and τ denotes a predefined threshold. The rth percentile of a data set refers to the value greater than or equal to $r\%$ of the data values. The PERCENTILE aggregate is much more robust than other statistic metrics such as MEAN, SUM, and MIN/MAX which can be easily manipulated by the data from a single or small set of dysfunctional or compromised nodes.

Our system is designed with three objectives. First, it should be *correct* in the sense that the monitoring service provider can decide when $f(\chi_r) > \tau$ happens with extremely low false positives and negatives. Second, it should be *communication-efficient* such that each node submits its data only when necessary. Last, it should be *privacy-preserving* in keeping individual users' data confidential.

This paper makes the following contributions.

- We are the first to motivate and formulate the problem of communication-efficient and privacy-preserving distributed stream monitoring for thresholded PERCENTILE aggregates to the best of our knowledge.

- We propose a novel technique for distributed stream monitoring of thresholded PERCENTILE aggregates with high communication efficiency and differential privacy guarantees. In our technique, the monitoring service provider constructs one or several safe (data) ranges based on the desired function $f(\cdot)$ and threshold τ. Each node can independently decide whether his statistic value in a new interval should be submitted based on the safe ranges and his statistic value in last interval. Powered by the differential privacy theory [5], our technique also ensures that each node's submitted data are not substantially different if one element of the node's data stream changes. Differential privacy guarantees can effectively prevent the monitoring service provider or any other internal/external adversary with arbitrary background knowledge from identifying the actual content of any particular data stream to breach the privacy of the corresponding node (user).
- We thoroughly evaluate the accuracy, communication efficiency, and privacy guarantees of our system through theoretical analysis and detailed simulation studies. Our results show that PriStream significantly reduces communication overhead and maintains differential privacy simultaneously.

The rest of this paper is organized as follows. Section II briefs the related work. Section III introduces the system and adversary models. Section IV outlines the background on differential privacy. Section V details our system design and analyzes its performance. Section VI evaluates our system through detailed MATLAB simulations based on both real-world and synthetic datasets. Section VII concludes this paper.

II. Related Work

A large chunk of work [1]–[4], [6]–[9] studies communication-efficient monitoring of distributed streams. A wide range of aggregate functions sought by the monitoring service provider have been covered, including SUM and COUNT [1], [2], inner products [6], and entropy [7], as well as MEAN and MIN/MAX in [1]. The work [8] aims to achieve efficient detection of distributed constraint violations. In addition, a novel geometric approach is proposed in [3] for monitoring threshold functions over distributed data streams. In this approach, a global monitoring task is decomposed into a set of geometric constraints applied locally in each node for deciding whether to submit the data. This geometric approach has been adopted by others for achieving various distributed stream monitoring goals [9]–[11]. Although elegant, these schemes cannot be applied to enable communication-efficient distributed stream monitoring of thresholded PERCENTILE aggregates.

Significant efforts have been made on privacy-preserving aggregation for distributed time-series data and/or providing differential privacy for individual data streams [12]–[22]. The PASTE algorithm in [12] targets historical time-series data and requires the pre-processing of all possible query results, so it cannot be applied for distributed real-time monitoring tasks. In addition, the algorithm in [13] enables an untrusted aggregator to compute the sum of distributed time-series data with differential privacy guarantees to all the data sources. This algorithm cannot be directly applied to distributed stream monitoring of thresholded PERCENTILE aggregates. Moreover, the framework in [4] enables monitoring arbitrary threshold functions over the MEAN aggregate of the statistics from distributed data-stream sources in a differentially privacy-preserving fashion. In addition, references [17]–[22] further offer fault tolerance against sensor failure. All these work focuses on additive aggregation and thus cannot be applied to our problem. In contrast, our PriStream system is the first work targeting differentially privacy-preserving distributed stream monitoring of thresholded PERCENTILE aggregates.

Privacy-preserving data aggregation is also studied in mobile sensing and wireless sensor networks [23], [24], [27]–[30]. The work [23], [24], [29] addresses privacy-preserving data aggregation by data slicing and mixing, but these schemes involve cooperation among peer nodes and does not apply to our scenario where sensor nodes work independently. Li *et al.* studied privacy-preserving MIN [28] and SUM [31] aggregations in mobile sensing systems, and these schemes cannot be applied to thresholded PERCENTILE aggregations. In addition, no differential privacy is guaranteed in [23], [28]–[30].

III. System and Adversary Models

A. System Model

We use a widely adopted model [1]–[4], [38]–[40] which consists of a service provider and k nodes denoted by $n_1, n_2, \cdots, n_k$. Affiliated with a human user or organization, each node n_i continuously performs the predetermined sensing task and can directly communicate with the service provider to submit data or receive instructions. In addition, unlike [25], [26], PriStream does not require communications or collaborations among the nodes.

We make the following assumptions for distributed stream monitoring of thresholded PERCENTILE aggregates. Time is divided into equal-length intervals, denoted by t_l for $l \in [1, \infty)$, and each node may generate new data items in each interval. Let $S_i = \{d_{i,1}, d_{i,2}, \cdots\}$ denote the data set of node n_i from the beginning, where $d_{i,l}$ for $l \in [1, \infty)$ refers to the lth data item in the data domain $\mathcal{D}$. In addition, we use $S_i(t_l) \subseteq S_i$ to denote the data items node n_i generated in interval t_l. In interval t_l, each node n_i can compute a statistic value decided by the service provider as $v_i(t_l) = g(S_i(t_l)) \in \mathcal{R}$, where $g(\cdot)$ is a publicly known function that generates statistic value based on the input data set. For example, $g(\cdot)$ can be the mean, average, count, or any other function.

The service provider aims to monitor whether the global condition $f(\chi_r(t_l)) > \tau$ holds in each interval j. Here $f(\cdot) : \mathcal{R} \rightarrow \mathcal{R}$ refers to an arbitrary single-parameter function chosen by the service provider; $\tau \in \mathcal{R}$ is the predetermined monitoring threshold; and $\chi_r(t_l)$ denotes the rth percentile of the statistic values from k sensor nodes. There is no universal definition for the rth percentile, and we adopt the nearest rank method for its simplicity. In particular, we first sort the k data values in the ascending order. $\chi_{r,j}$ is the smallest value in the list such that r percent of the data values is no larger than that value. More specifically, $\chi_r(t_l)$ is the value at position $\lceil rk/100 \rceil$ of the ordered list. Whenever the global condition is satisfied, the service provider takes corresponding actions such as broadcasting public safety alarms.

B. Adversary Model

The adversary can be internal to PriStream. An internal attacker can be the PriStream service provider, which is assumed to be honest-but-curious in the sense that it faithfully performs the system operations but is interested in the raw data of distributed nodes. This assumption is commonly adopted for system operators in the literature. An internal attacker can also be any distributed PriStream node which is curious about other nodes' raw data. In addition, the PriStream node can be honest by submitting real sensing data or malicious by reporting fake data. We assume that malicious PriStream nodes are the minority so that PriStream is always functional.

We also consider external attackers interested in the raw data of PriStream nodes to breach their privacy. An external attacker may compromise some PriStream nodes to be come internal attackers, but we assume that compromised nodes if any are the minority.

There can be collusion among internal attackers alone, external attackers alone, or internal and external attackers. We make a reasonable assumption that the number of attackers involved in a collusion is much smaller than the number of PriStream nodes which can be in thousands or more.

IV. Preliminaries on Differential Privacy

Differential privacy [5] is a recently proposed privacy model which guarantees strong privacy. It originally comes from the database discipline and has been applied in many other related areas [4], [36], [37]. In what follows, we first introduce the definition of ϵ-differential privacy and its properties. Then we outline two schemes to achieve ϵ-differential privacy.

Definition 1: **(*Adjacent Streams* [32]).** Two streams S_i and S'_i of n_i are defined as adjacent streams iff there exist $d, d' \in \mathcal{D}$ such that replacing d in S_i with d' will result in S'_i.

Definition 2: **(ϵ*-Differential Privacy* [5]).** A randomized algorithm Alg provides ϵ-differential privacy iff for any adjacent streams S_i and S'_i and any set O of possible outputs,

$$\mathbf{Pr}[\mathsf{Alg}(S_i) \in O] \leq \mathbf{Pr}[\mathsf{Alg}(S'_i) \in O] \times e^{\epsilon}, \tag{1}$$

where the probability is taken over the randomness of Alg.

The above definition means that a differentially private algorithm Alg will generate the same output over two streams with only one different element with almost the same probability. In general, ϵ is positive, and the smaller ϵ is, the stronger the differential privacy.

Definition 3: **(ℓ_ρ*-Sensitivity* [33]).** The ℓ_ρ-sensitivity of a function $g : S_i \to \mathcal{R}$ is defined as

$$\Delta_\rho(g) = \max_{S_i \approx S'_i} ||g(S_i) - g(S'_i)||_\rho, \tag{2}$$

where S_i and S'_i are two adjacent streams of node n_i which only differ in one element.

Composition properties. Differential privacy maintains a sequential composition property. In particular, a sequence of computations that each provides differential privacy independently also guarantee differential privacy, and the privacy cost of each computation is accumulated. For example, a sequential differentially private computation conducted by algorithms $\mathsf{Alg}_1, \mathsf{Alg}_2, \ldots, \mathsf{Alg}_n$, each with a privacy cost $\epsilon_1, \epsilon_2, \ldots, \epsilon_n$, respectively, can be processed as long as its privacy cost ϵ is greater or equal to $\sum_{i=1}^{n} \epsilon_i$.

Laplace [33] and exponential [34] mechanisms are commonly employed to achieve ϵ-differential privacy.

Definition 4: **(*Laplace Mechanism* [33]).** This mechanism is designed for real-valued outputs, and it directly adds noise drawn from a Laplace distribution to each original output value to achieve ϵ-differential privacy. More specifically, given a function $g : S \to \mathcal{R}$, the Laplace mechanism is defined as $g'(S) = g(S) + \mathsf{Laplace}(\Delta_1(g)/\epsilon)$, where $\mathsf{Laplace}(\lambda)$ for any λ denotes a Laplace distribution with probability density function $\mathbf{Pr}(x|\lambda) = \frac{1}{2\lambda} e^{\frac{-|x|}{\lambda}}$.

Definition 5: **(*Exponential Mechanism* [34])** This mechanism applies when target outputs are not real values or cannot be added with noises. An example is to sample one of several options while considering the desirability of each option. In particular, given a utility function $h : (\mathcal{D} \times O) \to \mathcal{R}$ which assigns a score to each output $r \in \mathcal{R}$, the exponential mechanism M which chooses an output $r \in \mathcal{R}$ based on a stream S_i of node n_i is defined as

$$\mathsf{M}(S_i, h) = \left\{ r \text{ with probability} \propto \exp\left(\frac{\epsilon h(S_i, r)}{2\Delta_1(h)}\right) \right\}. \tag{3}$$

V. PriStream Design

In this section, we elaborate on the design of PriStream.

A. Overview

The most intuitive method for monitoring whether global condition $f(\chi_r(t_l)) > \tau$ holds in every interval is to let each node n_i report its statistic value $v_i(t_l) = g(S_i(t_l))$ to the service provider, which can in turn decide $\chi_r(t_l)$ and test whether the global condition holds.

The above method has two obvious limitations. First, letting every node report its statistic value in every interval incurs significant communication overhead, especially if the reporting frequency need be sufficiently high (say, every five minutes) for real-time decision making. Since most sensor nodes in future smart cities are expected to have limited energy, their batteries will be quickly drained out and very difficult to replenish. Second, the service provider can learn the original data of all the nodes and thus violate the privacy of the corresponding users.

To address the first limitation, we propose a novel ranging technique to enable communication-efficient distributed monitoring. Specifically, we observe that testing whether the global condition $f(\chi_r(t_l)) > \tau$ holds does not require the service provider to know the actual value of $\chi_r(t_l)$. Instead, it suffices to know whether $\chi_r(t_l)$ falls into the range where $f(\chi_r(t_l)) > \tau$ holds. Recall that $\mathcal{R}$ is the domain of the statistic value at each node. It follows that $\chi_r(t_l) \in \mathcal{R}$ for every interval as well. Given function $f(\cdot)$ chosen by the service provider, we can divide $\mathcal{R}$ into a safe area $\mathcal{R}^+(t_l)$ and a unsafe area $\mathcal{R}^-(t_l)$, such that $\mathcal{R} = \mathcal{R}^+(t_l) \bigcup \mathcal{R}^-(t_l)$, $\mathcal{R}^+(t_l) \bigcap \mathcal{R}^-(t_l) = \emptyset$, and $f(\chi_r(t_l)) > \tau$ if and only if $\chi_r(t_l) \in \mathcal{R}^-(t_l)$. The global monitoring task can then be converted into testing whether $\chi_r(t_l) \in \mathcal{R}^-(t_l)$, which can be accomplished without knowing the actual value of $\chi_r(t_l)$ in every interval.

More specifically, for a given function $f(\cdot)$, the safe range $\mathcal{R}^+(t_l)$ and the unsafe range $\mathcal{R}^-(t_l)$ may each comprise multiple disjoint ranges. Without loss of generality, assume that $\mathcal{R}^+(t_l)$ and $\mathcal{R}^-(t_l)$ together comprise θ disjoint ranges $R_1(t_l), \ldots, R_\theta(t_l)$, where $\bigcup_{i=1}^{\theta} R_i(t_l) = \mathcal{R}$, $R_i(t_l) \bigcap R_j(t_l) = \emptyset$ for all $i \neq j$, and each $R_j(t_l)$ is either an open or closed range with left and right boundaries $\mathrm{l}_j(t_l)$ and $\mathrm{r}_j(t_l)$, respectively. Let $k_j(t_l)$ be the number of nodes with statistic values in R_j in interval t_l for all $j \in [1, \theta]$. It follows that $k = \sum_{j=1}^{\theta} k_j(t_l)$ for every interval $t_l = 1, 2, \ldots$. In every interval t_l, each node n_i reports to the service provider the index of range that its statistic value $v_i(t_l)$ falls into, which allows the service provider to compute $k_1(t_l), \ldots, k_\theta(t_l)$, and further determine which range $\chi_r(t_l)$ falls into and whether $f(\chi_r(t_l)) > \tau$ holds.

To tackle the second limitation, PriStream adopts the Laplace and the exponential mechanisms to provide differential privacy to participating nodes. Consider node n_i with statistic value $v_i(t_l) \in R_x(t_l)$ in interval t_l as an example. To ensure differential privacy, node n_i reports a perturbed interval index x' generated via a combination of the Laplace and the exponential mechanisms.

B. Detailed PriStream Operations

We now illustrate the detailed PriStream operations, which comprise initialization and communication-efficient phases.

1) Initialization Phase: The service provider starts the initialization phase at the end of interval t_l, where t_l refers to either the first interval t_1 or any subsequent interval (i.e., $l \geq 2$) for which the global monitoring condition has been changed in the preceding interval t_{l-1}. The purpose is to assign the range parameters to all nodes and collect each node's range index.

Operation Details

At the end of interval t_l, the service provider issues a system-wide query, which specifies the desired statistic metric generation function $g(\cdot)$, the precomputed disjoint ranges $\{R_j(t_l)\}_{j=1}^{\theta}$ with corresponding left and right boundaries $\{\langle \mathrm{l}_j(t_l), \mathrm{r}_j(t_l)\rangle\}_{j=1}^{\theta}$, the differential-privacy parameter ϵ.

Upon receiving the query, each node n_i for $\forall i \in [1, k]$ with its data vector $S_i(t_l)$ does the following in sequel.

1. Compute the desired statistic value $v_i(t_l) = g(S_i(t_l))$.
2. Find the real range index $x \in [1, \theta]$ such that $v_i(t_l) \in R_x(t_l)$.
3. Compute a perturbed range index $I_i(t_l)$ from x according to Alg. 1.
4. Send $I_i(t_l)$ to the service provider.

Algorithm 1: Generating Perturbed Range Index

```
input  : $\{(\mathrm{l}_i(t_l), \mathrm{r}_i(t_l))\}_{i=1}^{\theta}, \epsilon, v_i(t_l), \Delta_1(g)$
output : $I_i(t_l)$
Generate noise $\alpha_i \sim \mathsf{Laplace}\left(\frac{\Delta_1(g)}{\epsilon}\right)$ ;
for $j = 1, \ldots, \theta$ do
    Calculate $c_j(t_l) = \frac{\mathrm{l}_j(t_l) + \mathrm{r}_j(t_l)}{2}$ ;
    Generate a perturbed range $[\mathrm{l}_j(t_l) - \alpha_i, \mathrm{r}_j(t_l) + \alpha_i]$;
    if $v_i(t_l) < c_j(t_l)$ then
        $\mu_j(t_l) = \epsilon \cdot \frac{|c_j(t_l) - \mathrm{l}_j(t_l) + \alpha_i| - |c_j(t_l) - v_i(t_l)|}{2\Delta_1(g)}$;
    else
        $\mu_j(t_l) = \epsilon \cdot \frac{|r_j(t_l) - c_j(t_l) + \alpha_i| - |c_j(t_l) - v_i(t_l)|}{2\Delta_1(g)}$;
for $j = 1, \ldots, \theta$ do
    Calculate the probability that $v_i(t_l)$ locates in range $j$
    as $\hat{P}_j(t_l) = \frac{\exp(\mu_j(t_l))}{\sum_{j=1}^{\theta} \exp(\mu_j(t_l))}$;
Generate $u_i(t_l) \sim U[0, 1]$ ;
$\hat{P}_0 \leftarrow 0$ ;
for $j = 1, \ldots, \theta$ do
    if $\sum_{j'=0}^{j-1} \hat{P}_{j'}(t_l) \leq u_i(t_l) < \sum_{j'=0}^{j} \hat{P}_{j'}(t_l)$ then
        return $I_i(t_l) = j$;
```

After receiving all the perturbed range indexes, the service provider processes them as follows.

1. Count the number of perturbed range indexes being j as $k_j(t_l)$ for every $j \in [1, \theta]$.
2. Calculate the percentile value $P_j(t_l)$ at the range $R_j(t_l)$'s left boundary $\mathrm{l}_j(t_l)$ as $\sum_{i=j}^{\theta} k_i(t_l)/k$ for every $j \in [1, \theta]$.
3. Determine the range $R_x(t_l)$ in which $\chi_r(t_l)$ falls into by finding x such that $P_x > r/100 > P_{x+1}$.
4. Check whether $R_x(t_l)$ is a safe range. If not, take the predetermined action such as issuing a public safety alarm. Otherwise, keep silent. For both cases, proceed to communication-efficient phases.

2) Communication-Efficient Phase: In every interval $t_{l'}$ ($l' > l$), each node n_i computes a perturbed range index $I_i(t_l')$ according to Alg. 1 and reports $I_i(t_l')$ to the service provider only if $I_i(t_l') \neq I_i(t_{l'-1})$.

The detailed operations at each node n_i in communication-efficient phase are as follows.

1. Find the range $R_x(t_{l'})$ where $v_i(t_{l'})$ falls into.
2. Generate a perturbed range index $I_i(t_{l'})$ from the original range index x according to Alg. 1.
3. If $I_i(t_{l'}) \neq I_i(t_{l'-1})$, send $I_i(t_{l'})$ to the service provider and keep silent otherwise.

Upon receiving all the perturbed range indexes, the service provider processes them as follows.

1. For every node n_i that did not send $I_i(t_{l'})$, set $I_i(t_{l'}) = I_i(t_{l'-1})$.
2. Count the number of nodes with $I_i(t_{l'}) = j$ for every $j \in [1, \theta]$.
3. Calculate the percentile value $P_j(t_{l'})$ at the range $R_j(t_{l'})$'s left boundary $\mathrm{l}_j(t_{l'})$ as $\sum_{i=j}^{\theta} k_j(t_{l'})/k$ for every $j \in [1, \theta]$.
4. Determine the range $R_x(t_{l'})$ in which $\chi_r(t_{l'})$ falls into by finding x such that $P_x(t_{l'}) > r/100 > P_{x+1}(t_{l'})$.
5. Check whether $R_x(t_{l'})$ is a safe range. If not, take the predetermined action such as issuing a public safety alarm and keep silent otherwise.
6. Continue with the communication-efficient phase or start another initialization phase by broadcasting a new system-wide query in the next interval.

C. Performance Analysis

1) Correctness: The correctness of PriStream is affected by both the correctness of our proposed communication-efficient scheme and the accuracy guarantee after adopting mechanisms for differential privacy provision.

We first consider the correctness of our proposed scheme while ignoring the provision of differential privacy.

Theorem 1: Let $k_j(t_l)$ be the number of nodes with statistic value in range $R_j(t_l)$ for all $j \in [1, \theta]$ and $\chi_r(t_l)$ be the rth percentile of a set of statistic values $\{v_i(t_l)\}_{i=1}^{k}$. The global condition $f(\chi_r(t_l)) > \tau$ holds at time interval t_l for some predefined threshold τ if there exists an unsafe range $R_j(t_l)$ such that $P_j(t_l) > r/100 > P_{j+1}(t_l)$, where $P_x(t_l) = \sum_{i=x}^{\theta} k_i(t_l)/k$ for all $x \in [1, \theta]$.

Proof: Recall that $P_j(t_l) = \sum_{i=j}^{\theta} k_i(t_l)/k$ for all $j \in [1, \theta]$, where $k_j(t_l)$ is the number of nodes with statistic values in $R_j(t_l)$. Since $P_j(t_l) > r/100 > P_{j+1}(t_l)$, we have that $\sum_{i=j}^{\theta} k_i(t_l)/k > r/100 > \sum_{i=j+1}^{\theta} k_i(t_l)/k$. It follows that $\chi_r(t_l)$ is between the $\sum_{i=j+1}^{\theta} k_i(t_l)$th and the $\sum_{i=j}^{\theta} k_i(t_l)$th largest numbers among $\{v_i(t_l)\}_{i=1}^{k}$. We therefore have $\chi_r(t_l) \in R_j(t_l)$. Since $R_j(t_l)$ is an unsafe region, by definition, we have $f(\chi_r(t_l)) > \tau$. ■

Next, we consider the accuracy of PriStream after each node perturbs its range index using Alg. 1. Specifically, the accuracy of PriStream depends on how accurate the service provider can learn the number of values in each range in each interval, which in turn depends on how accurate Alg. 1 perturbs a range index. The following theorem guarantees that the perturbed range index output by Alg. 1 would not be very different from the range index before perturbation.

Theorem 2: If node n_i's statistic value $v_i(t_l)$ is outside of range $R_j(t_l)$ and the distance between $v_i(t_l)$ and the closer boundary of $R_j(t_l)$ is at least $\frac{2\Delta_1(g)}{\epsilon} \log \frac{1-\delta}{\delta^{1.5}(\theta-1)}$, then Alg. 1 will output a perturbed range index $I_i(t_l) = j$ with probability at most 2δ.

Proof: In time interval t_l, the error introduced in either initialization or communication-efficient phase comes from the perturbation of the range with Laplace noise α_i and the exponential mechanism. For the range perturbation operation, since α_i is sampled from a Laplace distribution $\mathsf{Laplace}\left(\frac{\Delta_1(g)}{\epsilon}\right)$ with the cumulative distribution function

$$F(x) = \begin{cases} \frac{1}{2} \exp\left(\frac{x\epsilon}{\Delta_1(g)}\right), & \text{if } x < 0, \\ 1 - \frac{1}{2} \exp\left(\frac{-x\epsilon}{\Delta_1(g)}\right), & \text{if } x \geq 0. \end{cases} \tag{4}$$

Assume that $|\alpha_i|$ is at most ψ with probability $1 - \delta$, where $\psi > 0$. We have

$$1 - 2 \cdot \frac{1}{2} \exp\left(-\frac{\psi\epsilon}{\Delta_1(g)}\right) = 1 - \delta.$$

Solving the above equation, we have $\psi = \frac{\Delta_1(g)}{\epsilon} \log \frac{1}{\delta}$. Therefore, we know that

$$\mathbf{Pr}\left[|\alpha_i| \leq \frac{\Delta_1(g)}{\epsilon} \log \frac{1}{\delta}\right] \geq 1 - \delta. \tag{5}$$

In addition, assume that with probability $1 - \delta$, the statistic value $v_i(t_l)$ at a node n_i exceeds a range boundary by φ. We further have

$$1 - \frac{\exp(-\frac{\epsilon\varphi}{2\Delta_1(g)})}{(\theta - 1)\exp(0) + \exp(-\frac{\epsilon\varphi}{2\Delta_1(g)})} = 1 - \delta.$$

Solving the above equation, we obtain

$$\varphi = \frac{2\Delta_1(g)}{\epsilon} \log \frac{1-\delta}{\delta(\theta-1)},$$

and

$$\mathbf{Pr}\left[\varphi \geq \frac{2\Delta_1(g)}{\epsilon} \log \frac{1-\delta}{\delta(\theta-1)}\right] \geq 1 - \delta. \tag{6}$$

Considering the above two factors simultaneously, if the distance between $v_i(t_l)$ and $R_j(t_l)$ is larger than $\frac{2\Delta_1(g)}{\epsilon} \log \frac{1-\delta}{\delta^{1.5}(\theta-1)}$, Alg. 1 will output a perturbed range index $I_i(t_l) = j$ with probability at most 2δ, which can also be written as

$$\mathbf{Pr}\left[\varphi \leq \frac{2\Delta_1(g)}{\epsilon} \log \frac{1-\delta}{\delta^{1.5}(\theta-1)}\right] \leq 2\delta. \tag{7}$$

■

Theorem 3: If node n_i's statistic value $v_i(t_l)$ is inside the range R_j and the distance between $v_i(t_l)$ and the closer boundary of R_j is more than $\frac{2\Delta_1(g)}{\epsilon} \log \frac{1-\delta}{\delta^{1.5}(\theta-1)}$, then Alg. 1 will output a perturbed range index $I_i(t_l) = j$ with probability at least $(1 - 2\delta)$.

The proof of Theorem 3 is similar to that of Theorem 2.

2) Communication Overhead: The following theorem gives the communication overhead incurred by PriStream.

Theorem 4: Given a PriStream execution process with a initialization and b communication-efficient phases, PriStream incurs the communication overhead of $a(|\mathcal{M}| + \varpi k) + \varpi \sum_{l=1}^{b} \lambda_l$, where $|\mathcal{M}|$ is the communication overhead incurred by broadcasting system information, $\varpi = \lfloor \log_2(\theta - 1) \rfloor + 1$ is the size of a range index, k is the number of nodes, λ_l

is the number of nodes that submit range index to the service provider in lth communication-efficient phase, and $l \in [1, b]$.

Proof: We analyze the communication overhead incurred in initialization and communication-efficient phases separately.

In the initialization phase, the service provider need send a message $\mathcal{M}$ containing the desired statistic metric generation function $g(\cdot)$, the precomputed range parameters $\{R_j(t_l)\}_{j=1}^{\theta}$, the differential-privacy parameter ϵ for differential-privacy mechanism. We denote the communication overhead incurred by transmitting $\mathcal{M}$ by $|\mathcal{M}|$. In addition, each node sends a ϖ-bit range index to the service provider, totaling to ϖk bits.

In every communication-efficient phase, the node whose perturbed range index is different from that in last interval sends a ϖ-bit range index to the service provider, which incurs total communication overhead of $\lambda_l \varpi$ bits, where $\lambda_l \in [1, k]$ is the number of nodes that submit range index to the service provider in lth communication-efficient phase, where $l \in [1, b]$.

In summary, the overall communication overhead incurred by a PriStream execution process with a initialization and b communication-efficient phases is given by $a(|\mathcal{M}| + \varpi k) + \varpi \sum_{l=1}^{b} \lambda_l$. ■

3) Privacy Analysis: The privacy of our proposed scheme is guaranteed by the following theorem.

***Theorem** 5:* PriStream consisting of a initialization and b communication-efficient phases maintains $2(a + b)\epsilon$-differential privacy.

Proof: We follow the proof technique in [32] to prove that PriStream guarantees ϵ-differential privacy for each participating node in each phase. We consider all the noise components added in our scheme to obtain privacy guarantees for a multi-round process. For each node n_i, given two adjacent data streams $S_i(t_l)$ and $S'_i(t_l)$ that differ in only one element, we consider the scheme execution process as a sequential process consisting of a initialization and b communication-efficient phases.

In what follows, we analyze each phase of the PriStream execution in detail to show how different operations on $S_i(t_l)$ and $S'_i(t_l)$ will lead to the same output. For convenience, we use E and E' to denote the PriStream execution process over two adjacent data streams $S_i(t_l)$ and $S'_i(t_l)$, respectively.

Initialization phase. Assume that each node n_i generates its statistic value as $v_i(t_l) = g(S_i(t_l))$ and $v'_i(t_l) = g(S'_i(t_l))$ in two executions E and E', respectively. For execution E, each node n_i generates Laplace noise α_i from a Laplace distribution. For execution E', the Laplace noise generated by node n_i is α_i. In the initialization phase, the execution E outputs range index j with probability $\hat{P}_j(t_l)$, which is given by

$$\hat{P}_j(t_l) = \frac{\exp(\mu_j(t_l))}{\sum_{j=1}^{\theta} \exp(\mu_j(t_l))}, \tag{8}$$

where

$$\mu_j(t_l) = \epsilon \cdot \frac{h}{2\Delta_1(g)}, \tag{9}$$

$$h = \begin{cases} |c_j(t_l) - \mathrm{l}_j(t_l) + \alpha_i| - |c_j(t_l) - v_i(t_l)|, \\ \qquad\qquad \forall\ v_i(t_l) < c_j(t_l), 1 \le j \le \theta, \\ |\mathrm{r}_j(t_l) - c_j(t_l) + \alpha_i| - |c_j(t_l) - v_i(t_l)|, \\ \qquad\qquad \forall\ v_i(t_l) \ge c_j(t_l), 1 \le j \le \theta, \end{cases} \tag{10}$$

$c_j(t_l) = \frac{\mathrm{l}_j(t_l) + \mathrm{r}_j(t_l)}{2}$, $\mathrm{l}_j(t_l)$ and $\mathrm{r}_j(t_l)$ are the left and right boundaries of the jth range in which $v_i(t_l)$ actually locates in time interval t_l, respectively. The ℓ_1-sensitivity of utility function h is $\Delta_1(g)$.

Assume $v_i(t_l) \in R_j(t_l)$ in time interval t_l, we have

$$\begin{aligned} \frac{\Pr[\mathsf{M}(S_i(t_l), h) = j]}{\Pr[\mathsf{M}(S'_i(t_l), h) = j]} &= \frac{\left(\frac{\exp(\mu_j(t_l))}{\sum_{x=1}^{\theta} \exp(\mu_x(t_l))}\right)}{\left(\frac{\exp(\mu'_j(t_l))}{\sum_{x=1}^{\theta} \exp(\mu'_x(t_l))}\right)} \\ &= \left(\frac{\exp(\mu_j(t_l))}{\exp(\mu'_j(t_l))}\right) \cdot \left(\frac{\sum_{x=1}^{\theta} \exp(\mu'_x(t_l))}{\sum_{x=1}^{\theta} \exp(\mu_x(t_l))}\right) \\ &= \exp\left(\mu_j(t_l) - \mu'_j(t_l)\right) \cdot \left(\frac{\sum_{x=1}^{\theta} \exp(\mu'_x(t_l))}{\sum_{x=1}^{\theta} \exp(\mu_x(t_l))}\right) \end{aligned} \tag{11}$$

For $v_i(t_l) < c_j(t_l)$, we have

$$\begin{aligned} \mu_j(t_l) - \mu'_j(t_l) =& \epsilon \cdot \frac{|c_j(t_l) - \mathrm{l}_j(t_l) + \alpha_i| - |c_j(t_l) - v_i(t_l)|}{2\Delta_1(g)} \\ & - \epsilon \cdot \frac{|c_j(t_l) - \mathrm{l}_j(t_l) + \alpha'_i| - |c_j(t_l) - v'_i(t_l)|}{2\Delta_1(g)} \\ \le& \frac{\epsilon}{2} \cdot \frac{|\alpha_i - \alpha'_i| + |v_i(t_l) - v'_i(t_l)|}{\Delta_1(g)} \\ \le& \epsilon, \end{aligned} \tag{12}$$

$$\begin{aligned} \mu'_j(t_l) &= \epsilon \cdot \frac{|c_j(t_l) - \mathrm{l}_j(t_l) + \alpha'_i| - |c_j(t_l) - v'_i(t_l)|}{2\Delta_1(g)} \\ &\le \epsilon \cdot \frac{|c_j(t_l) - \mathrm{l}_j(t_l) + \alpha_i|}{2\Delta_1(g)} \\ &\quad - \epsilon \cdot \frac{-|c_j(t_l) - v_i(t_l)| - \Delta_1(g)}{2\Delta_1(g)} \\ &= \epsilon \cdot \frac{|c_j(t_l) - \mathrm{l}_j(t_l) + \alpha_i| - |c_j(t_l) - v_i(t_l)| + 2\Delta_1(g)}{2\Delta_1(g)} \\ &= \mu_j(t_l) + \epsilon. \end{aligned} \tag{13}$$

For $v_i(t_l) \ge c_j(t_l)$, we have

$$\begin{aligned} \mu_j(t_l) - \mu'_j(t_l) =& \epsilon \cdot \frac{|\mathrm{r}_j(t_l) - c_j(t_l) + \alpha_i| - |c_j(t_l) - v_i(t_l)|}{2\Delta_1(g)} \\ & - \epsilon \cdot \frac{|\mathrm{r}_j(t_l) - c_j(t_l) + \alpha'_i| - |c_j(t_l) - v'_i(t_l)|}{2\Delta_1(g)} \\ \le& \epsilon \cdot \frac{|\alpha_i - \alpha'_i| + |v_i(t_l) - v'_i(t_l)|}{2\Delta_1(g)} \\ \le& \epsilon, \end{aligned} \tag{14}$$

$$
\begin{aligned}
\mu_j'(t_l) &= \epsilon \cdot \frac{|\mathrm{r}_j(t_l) - c_j(t_l) + \alpha_i'| - |c_j(t_l) - v_i'(t_l)|}{2\Delta_1(g)} \\
&\leq \epsilon \Bigg(\cdot \frac{|\mathrm{r}_j(t_l) - c_j(t_l) + \alpha_i| + \Delta_1(g)}{2\Delta_1(g)} \\
&\quad + \cdot \frac{-|c_j(t_l) - v_i(t_l)| + \Delta_1(g)}{2\Delta_1(g)} \Bigg) \\
&= \epsilon \cdot \frac{|\mathrm{r}_j(t_l) - c_j(t_l) + \alpha_i| - |c_j(t_l) - v_i(t_l)| + 2\Delta_1(g)}{2\Delta_1(g)} \\
&= \mu_j(t_l) + \epsilon.
\end{aligned}
\tag{15}
$$

Therefore, we have

$$
\begin{aligned}
\frac{\Pr[\mathsf{M}(S_i(t_l), h) = j]}{\Pr[\mathsf{M}(S_i'(t_l), h) = j]} &\leq \exp(\epsilon) \left(\frac{\sum_{x=1}^{\theta} \exp\left(\mu_x(t_l) + \epsilon\right)}{\sum_{x=1}^{\theta} \exp(\mu_x(t_l))} \right) \\
&= \exp(\epsilon) \left(\frac{\sum_{x=1}^{\theta} \exp(\mu_x(t_l)) \cdot \exp(\epsilon)}{\sum_{x=1}^{\theta} \exp(\mu_x(t_l))} \right) \\
&= \exp(2\epsilon),
\end{aligned}
\tag{16}
$$

which indicates that the initialization phase guarantees 2ϵ-differential privacy.

Communication-efficient phase. The operations of each communication-efficient phase is similar to those of the initialization phase except for the case in which some nodes do not need to send the index of the range in which it resides to the service provider if the statistic value remains in the same range as that in the previous phase. Therefore, we can similarly obtain the same result that each communication-efficient phase guarantees 2ϵ-differential privacy.

The whole PriStream execution process. As mentioned, the whole execution process is a sequential process consisting of a initialization and b communication-efficient phases. Since the noise added in each phase is drawn independently, the probability difference of obtaining the same output for the whole execution process can be considered the multiplication of the probability difference in each phase. Therefore, for the whole execution process, the probability of obtaining the output based on execution E' differs from that of execution E by a factor of at most $\exp(2(a+b)\epsilon)$, which guarantees $2(a+b)\epsilon$-differential privacy. ■

VI. Performance Evaluation

In this section, we evaluate the performance of PriStream via MATLAB simulations based on both real-world and synthetic datasets.

A. Simulation Setup

We adopt the following metrics to evaluate the performance of PriStream.

- *Communication overhead*: We quantify the communication overhead by the number of bits transmitted between the service provider and nodes during stream monitoring.

TABLE I
Default Simulation Settings

Para.	**Value**	**Meaning**
k	1000	The number of nodes
ϵ	0.15	The differential privacy parameter
θ	100	The number of ranges
r	80	The percentile value

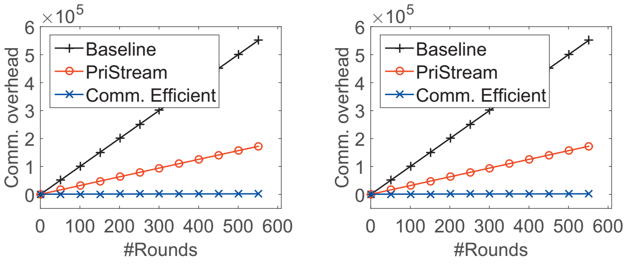
- *Accuracy*: The accuracy is used to evaluate the utility after introducing Alg. 1. We treat the range index of each round in communication-efficient scheme as the ground truth and compare it with the range index of the corresponding round in PriStream. The accuracy is defined as the ratio of the number of the same indexes over the total rounds.
- *Privacy loss*: The privacy loss is defined as

$$
\hat{\epsilon} = \max \ln \frac{\mathbf{Pr}[\mathsf{M}(S_i(t_l), h) = j]}{\mathbf{Pr}[\mathsf{M}(S_i'(t_l), h) = j]},
\tag{17}
$$

  where S_i and S_i' are two adjacent streams which differ in only one element. Obviously, the smaller $\hat{\epsilon}$, the less impact of the change of one element on the range index generation algorithm, the higher level of differential privacy is offered, and vice versa.

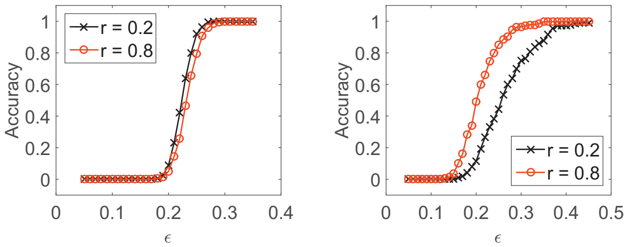
We use two datasets to evaluate the performance of PriStream. The first dataset is MHEALTH [35], a mobile health dataset that comprises body motion and vital sign measures for several volunteers of diverse profiles while performing 12 physical activities such as walking, running and climbing stairs. The dataset contains totally 1,215,745 recordings, each of which is composed of 24 types of signals from the sensors such as accelerometer, gyroscope, and magnetometer. In this paper, we used all the 1,215,745 recordings for one type of signal because they are at the same scale and are the focus of this paper; we leave the monitoring and evaluation of multi-dimension streams with different scales as future work. We then randomly partition them into 1000 subsets, representing 1000 distributed nodes, each of which has about 1216 data items, corresponding to 1216 intervals. The service provider starts the initialization phase at interval 608 and then conducts subsequent 608 rounds of queries, and each node will generate $v_i(t_l)$ based on its previous 608 data items. The second dataset is a synthetic dataset generated by MATLAB used to simulate the case with different data distribution. In particular, the data in MHEALTH dataset follow Gaussian distribution. We extract the data range from MHEALTH dataset and then generate a synthetic dataset which is uniformly distributed in the same data range from. All other metrics such as the number of nodes, the number of intervals, and the number of query rounds in synthetic dataset are the same as that in MHEALTH dataset.

The default simulation settings are summarized in Tab. I.



(a) MHEALTH dataset (b) Synthetic dataset

Fig. 1. Impact of the number of rounds on communication overhead.



(a) MHEALTH dataset (b) Synthetic dataset

Fig. 2. Impact of ϵ on accuracy.

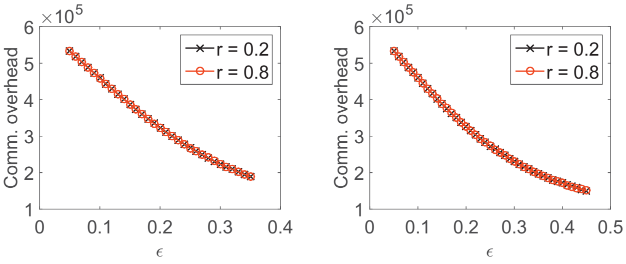
B. Simulation Results

We report the simulation results of a communication-efficient scheme (e.g., PriStream without Alg. 1), PriStream and a baseline scheme that lets each node directly submit its statistic value to the service provider.

Fig. 1 compares the communication overhead of the baseline, the communication-efficient, and PriStream schemes with b (i.e., the number of rounds) varying from 1 to 600. We can see that the communication-efficient and PriStream schemes incur much lower communication overhead than the baseline scheme does. The reason is than the number of nodes that submit data to the service provider in communication-efficient and PriStream schemes is much smaller than in the baseline scheme. Besides, by reporting range index instead of statistic value, communication overhead is further reduced. In addition, we can see that PriStream scheme incurs higher communication overhead than that of the communication-efficient scheme. The reason is that the range index of each node's statistic value is perturbed to other range indexes for the protection of data privacy, resulting in more range index being submitted to the service provider.

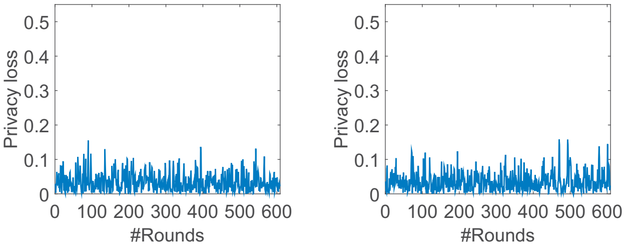
Fig. 2 shows the relationship between the accuracy and the differential privacy parameter ϵ. Obviously, the baseline scheme, which does not consider the data privacy, achieves 100% accuracy without being affected by the change of ϵ. However, the accuracy of PriStream increases as the differential privacy parameter ϵ increases. The reason is that as ϵ increases, the perturbed range index generated by Alg. 1 is more likely to be the same as its perturbed range index in previous interval, leading to less range index updates.

Fig. 3 shows the impact of differential privacy parameter ϵ on the communication overhead of PriStream. We can see that the communication overhead decreases as ϵ increases, demonstrating a trade-off between ϵ and communication overhead. The reason is that the higher ϵ, the higher the probability that a node's statistic value remains in the same range after perturbation, and the fewer nodes that need to report range index updates.



(a) MHEALTH dataset (b) Synthetic dataset

Fig. 3. Impact of ϵ on communication overhead.



(a) MHEALTH dataset (b) Synthetic dataset

Fig. 4. Privacy loss of PriStream.

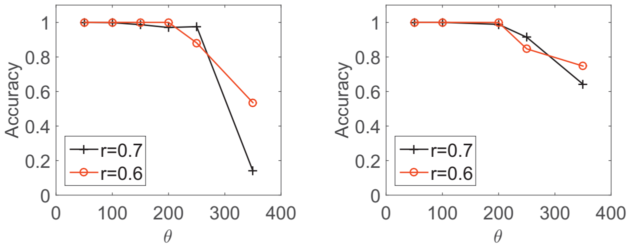
Fig. 4 illustrates the privacy loss after using PriStream. We can see that the privacy loss is always below 0.15. According to Tab. I, the simulation setting of differential privacy parameter is $\epsilon = 0.15$, which indicates that our designed scheme can always guarantee the desired 2ϵ-differential privacy.

Fig. 5 shows the impact of the number of ranges on accuracy. We can easily find that the accuracy decreases as the number of ranges increases for both datasets. The reason is that the larger the number of ranges, the smaller the range size, the higher the probability that the statistic value is perturbed to other ranges.

Fig. 6 shows the impact of θ on communication overhead. We can see that the communication overhead increases as θ increases. The reason is that the larger the θ is, the smaller the range size, the higher the probability that the statistic value is perturbed to a different range, and the higher communication overhead.

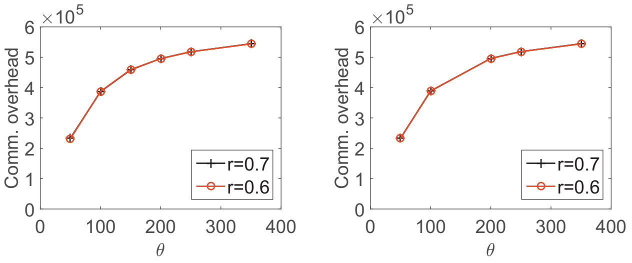
VII. Conclusion

This paper proposes PriStream, a novel privacy-preserving and communication-efficient distributed stream monitoring system. Different from previous work on monitoring the function of mean statistic value, our proposed scheme monitors the statistic value at the given percentile rank in a privacy-preserving and communication-efficient fashion. The efficacy and efficiency of our PriStream are confirmed by detailed MATLAB simulations.



(a) MHEALTH dataset (b) Synthetic dataset

Fig. 5. Impact of θ on accuracy.



(a) MHEALTH dataset (b) Synthetic dataset

Fig. 6. Impact of θ on communication overhead.

VIII. Acknowledgement

This work was supported in part by the US National Science Foundation under grants CNS-1514381, CNS-1421999, CNS-1320906, CNS-1422301, and CNS-1514014.

References

[1] C. Olston, J. Jiang, and J. Widom, "Adaptive filters for continuous queries over distributed data streams," in *SIGMOD'03*, San Diego, CA, USA, Jun. 2003.

[2] R. Keralapura, G. Cormode, and J. Ramamirtham, "Communication-efficient distributed monitoring of thresholded counts," in *SIGMOD'06*, Chicago, IL, USA, Jun. 2006.

[3] I. Sharfman, A. Schuster, and D. Keren, "A geometric approach to monitoring threshold functions over distributed data streams," in *SIGMOD'06*, Chicago, IL, USA, Jun. 2006.

[4] A. Friedman, I. Sharfman, D. Keren, and A. Schuster, "Privacy-preserving distributed stream monitoring," in *NDSS'14*, San Diego, CA, USA, Feb. 2014.

[5] C. Dwork, "Differential privacy," in *ICALP'06*, Venice, Italy, Jul. 2006.

[6] G. Cormode and M. Garofalakis, "Approximate continuous querying over distributed streams," *ACM Trans.Database Syst.*, vol. 33, no. 2, pp. 9:1–9:39, Jun. 2008.

[7] C. Arackaparambil, J. Brody, and A. Chakrabarti, "Functional monitoring without monotonicity," in *ICALP'09*, Rhodes, Greece, Jul. 2009.

[8] S. Agrawal, S. Deb, K. Naidu, and R. Rastogi, "Efficient detection of distributed constraint violations," in *ICDE'07*, Delhi, India, April 2007.

[9] M. Garofalakis, D. Keren, and V. Samoladas, "Sketch-based geometric monitoring of distributed stream queries," *Proc. VLDB Endow.*, vol. 6, no. 10, pp. 937–948, Aug. 2013.

[10] D. Keren, G. Sagy, A. Abboud, D. Ben-David, A. Schuster, I. Sharfman, and A. Deligiannakis, "Geometric monitoring of heterogeneous streams," *IEEE TKDE*, vol. 26, no. 8, pp. 1890–1903, Jul. 2014.

[11] A. Lazerson, I. Sharfman, D. Keren, A. Schuster, M. Garofalakis, and V. Samoladas, "Monitoring distributed streams using convex decompositions," *Proc. VLDB Endow.*, vol. 8, no. 5, pp. 545–556, Jan. 2015.

[12] V. Rastogi and S. Nath, "Differentially private aggregation of distributed time-series with transformation and encryption," in *SIGMOD'10*, Indianapolis, IN, June 2010.

[13] E. Shi, T.-H. Chan, E. FxPal, R. Chow, and D. Song, "Privacy-preserving aggregation of time-series data," in *NDSS'11*, San Diego, CA, Feb. 2011.

[14] Q. Li and G. Cao, "Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error," in *PET*, vol. 7981, 2013, pp. 60–81.

[15] F. Eigner, A. Kate, M. Maffei, F. Pampaloni, and I. Pryvalov, "Differentially private data aggregation with optimal utility," in *ACSAC'14*, New Orleans, LA, 2014, pp. 316–325.

[16] R. Chen, A. Reznichenko, P. Francis, and J. Gehrke, "Towards statistical queries over distributed private user data," in *NSDI'12*, San Jose, CA, 2012.

[17] G. Acs and C. Castelluccia, "I have a dream! (differentially private smart metering)," in *IH*, Prague, Czech Republic, 2011, pp. 118–132.

[18] T.-H. Chan, E. Shi, and D. Song, "Privacy-preserving stream aggregation with fault tolerance," in *FC*, 2012, pp. 200–214.

[19] M. Jawurek and F. Kerschbaum, "Fault-tolerant privacy-preserving statistics," in *PET*, vol. 7384, 2012, pp. 221–238.

[20] M. Joye and B. Libert, "A scalable scheme for privacy-preserving aggregation of time-series data," in *FC*, vol. 7859, 2013, pp. 111–125.

[21] J. Won, C. Ma, D. Yau, and N. Rao, "Proactive fault-tolerant aggregation protocol for privacy-assured smart metering," in *INFOCOM'14*, April 2014, pp. 2804-2812.

[22] H. Bao and R. Lu, "A new differentially private data aggregation with fault tolerance for smart grid communications," *IEEE IoT Journal*, vol. 2, no. 3, pp. 248–258, June 2015.

[23] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: privacy-preserving data aggregation in people-centric urban sensing systems," in *INFOCOM'10*, San Diego, CA, Mar. 2010.

[24] R. Zhang, J. Shi, Y. Zhang, and C. Zhang, "Verifiable privacy-preserving aggregation in people-centric urban sensing systems," in *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp.268–278, Sep. 2013.

[25] J. Sun, R. Zhang, and Y. Zhang, "Privacy-preserving spatiotemporal matching," in *INFOCOM'13*, Turlin, Italy, Apr. 2013.

[26] J. Sun, X. Chen, J. Zhang, Y. Zhang, J. Zhang, "SYNERGY: A game-theoretical approach for cooperative key generation in wireless networks," in *INFOCOM'14*, Toronto, Canada, Apr. 2014.

[27] Q. Li and G. Cao, "Efficient privacy-preserving stream aggregation in mobile sensing with low aggregation error," in *PETS'12*, Vigo, Spain, July 2012.

[28] ——, "Efficient and privacy-preserving data aggregation in mobile sensing," in *ICNP'12*, Austin, TX, Oct. 2012.

[29] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. F. Abdelzaher, "PDA: Privacy-preserving data aggregation in wireless sensor networks," in *INFOCOM'07*, Anchorage, Alaska, May 2007, pp. 2045–2053.

[30] T. Jung, X. Mao, X.-Y. Li, S.-J. Tang, W. Gong, and L. Zhang, "Privacy-preserving data aggregation without secure channel: Multivariate polynomial evaluation," in *ICNP'13*, Turin, Italy, Apr. 2013.

[31] H. Liu, S. Saroiu, A. Wolman, and H. Raj, "Software abstractions for trusted sensors," in *MobiSys'12*, Low Wood Bay, Lake District, UK, June 2012, pp. 365–378.

[32] C. Dwork, M. Naor, T. Pitassi, and G. Rothblum, "Differential privacy under continual observation," in *STOC'10*, Cambridge, Ma, Jun. 2010.

[33] C. Dwork, F. McSherry, K. Nissim, and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *TCC'06*, New York, NY, Mar. 2006.

[34] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *FOCS'07*, Providence, RI, Oct. 2007.

[35] O. Banos, R. Garcia, J. Holgado, M.Damas, H. Pomares, I. Rojas, A. Saez, and C. Villalonga, "mhealthdroid: a novel framework for agile development of mobile health applications," in *IWAAL'14*, Belfast, Northern Ireland, Dec. 2014.

[36] Z. Zhang, H. Zhang, S. He, and P. Cheng, "Achieving bilateral utility maximization and location privacy preservation in database-driven cognitive radio networks," in *MASS'15*, Dallas, TX, Oct. 2015.

[37] X. Jin and Y. Zhang, "Privacy-preserving crowdsourced spectrum sensing," in *INFOCOM'16*, San Francisco, CA, Apr. 2016.

[38] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: platform and applications," in *HotWeb'15*, Washington D.C., Nov. 2015.

[39] J. Sun, R. Zhang, X. Jin, and Y. Zhang, "SecureFind: Secure and Privacy-Preserving Object Finding via Mobile Crowdsourcing," in *IEEE Trans. Wireless Commun.*, vol. pp, no. 99, Oct. 2015.

[40] R. Zhang, J. Sun, Y. Zhang, and C. Zhang, "Secure spatial top-k query processing via untrusted location-based service provider," in *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 1, pp. 111-124. Jan./Feb. 2015.