A Novel Temporal Perturbation Based Privacy-preserving Scheme for Real-time Monitoring Systems

Xinyu Yang[a], Xuebin Ren[a], Shusen Yang[b], Julie McCann[c]

[a] *Dept. of Computer Science and Technology, Xi'an Jiaotong University, China.*
[b] *Corresponding Author, Department Electrical Engineering & Electronics, University of Liverpool, UK.*
[c] *Department of Computing, Imperial College London, UK.*

Abstract

In real-time monitoring systems, participant's privacy could be easily exposed when the time-series of sensing measurements are obtained accurately by adversaries. To address privacy issues, a number of privacy-preserving schemes have been designed for various monitoring applications. However, these schemes either lack considerations for temporal privacy or have less resistance to filtering attacks, or cause time delay with low utility. In this paper, we introduce a lightweight temporal perturbation based scheme, where sensor readings are buffered and disordered to obfuscate the temporal information of the original sensor measurement stream with differential privacy. Besides, we design the operations on the system server side to exploit the data utility in measurements from large number of sensors. We evaluate the performance of the proposed scheme through both rigorous theoretical analysis and extensive simulation experiments in comparison with related existing schemes. Evaluation results show that the proposed scheme manages to preserve both the temporal privacy and measurement privacy with filter-resistance, and achieves better performance in terms of computational overhead, data utility of real-time aggregation, and individual accumulation.

Keywords: Real-time monitoring system, privacy-preserving, temporal privacy, temporal perturbation

1. Introduction

With the development of smart devices embedded with sensors, real-time monitoring systems (e.g., participatory sensing [1] and smart metering systems [2]) have attracted more and more attention again[3][4][5]. A real-time

Email addresses: yxyphd@mail.xjtu.edu.cn (Xinyu Yang), xb.ren@stu.xjtu.edu.cn (Xuebin Ren), shusen.yang@liverpool.ac.uk (Shusen Yang), j.mccann@imperial.ac.uk (Julie McCann)

monitoring system consists of three entities: sensor meters, the system server and the third-party users, as shown in Fig. 1. The system server aggregates real-time measurements from sensors in both time and user dimensions, i.e, *Real-time aggregation* and *off-line accumulation.* Real-time aggregation of different users' measurements at one time, can provide real-time analysis of crowded dynamics; While, off-line accumulation for individuals in a period of time can provide accurate individual awareness. For example, in the smart metering system [2], a typical real-time monitoring system, on one hand, the utility supplier collects consumers' real-time consumptions from smart meters to monitor the states of the grid and guide energy scheduling; on the other hand, the periodic consumptions of individual consumers will be accumulated for consumption awareness and billing for consumers.

Once sent out, users' measurements could be shared with system server and third-party users in the real-time monitoring systems. Both high timeliness and fine granularity in real-time measurements can threaten users' privacy. Especially, user's behavioural privacy could be compromised from the fine-grained measurements collected from sensors [1][2][6][7]. For example, electricity usages in the smart metering system can be used to infer the ON/OFF states of users' appliances [8] and the continuous accelerator readings can leak the physical conditions of individuals [4]. Generally, privacy does not equal to the confidentiality. The latter means the data isn't being used or accessed by unauthorized individuals and it can be basically secured against the eavesdroppers via encrypted channels between sensor meters and the application server. While, privacy means the appropriate use of data and it usually needs to confront with potential data analyst in the system, who is a legitimate receiver as well as a snooping adversary. The reference of basic end-to-end encryption technique is then flawed in the privacy domain [9]. Hence, it is desirable to design effective privacy preserving mechanisms for real-time monitoring systems.

To address privacy issues in real-time monitoring systems, a number of research efforts have been made in the past [10][11][12][5][13][14] [15][16][17][18] [19][20]. For example, Fan *et al.* [5] proposed an adaptive approach to aggregate real-time monitoring with differential privacy [21]. Kamat *et al.*[17] propose to delay the packets delivery to obfuscate the time information of events detected. Nonetheless, most of these privacy-preserving schemes are designed for specific applications and not applicable for generic real-time monitoring systems. Besides, existing schemes mainly focus on data privacy and barely consider the temporal privacy, which could leak the temporal context information. Even if the temporal privacy is considered, the scheme still lose the timeliness, which is necessary in the real-time monitoring systems. In addition, few schemes have considered to support multiple operations (e.g., both the real-time aggregation and off-line accumulation) with high utility-privacy tradeoffs. Hence, this calls for an effective privacy preserving scheme, which consider both the data privacy and temporal privacy, and can be flexible for various applications and operations with high utility-privacy tradeoffs.

In this paper, we present a novel temporal perturbation based privacy-preserving scheme for real-time monitoring systems, which can protect both the data privacy as well as temporal privacy and achieve high utility-privacy tradeoffs for both real-time aggregation and off-line accumulation with filter-resistance. Particularly, our scheme focuses on the temporal privacy and exploits temporal correlations in real-time sensor measurements to leverage the utility-privacy tradeoff. Due to the internal steadiness and continuity of the physical world [22] [23], measured data are temporal correlated, which contains much more information than our expectation. In our scheme, from a micro-perspective, the time-series measurements are perturbed symmetrically in terms of time to distort detailed features (i.e., privacy) of individual measurements. Besides, from the macro-perspective, residual temporal correlation can be utilized to release enough data utility (or fidelity) for aggregation and accumulation on the system server side. Our contributions can be summarized as follows:

- **Temporal perturbation in real-time monitoring systems:** We build a generic mechanism to apply temporal perturbation to preserve both the temporal and measurement privacy in the real-time monitoring systems, which is different from the existing temporal perturbation based schemes with high time delay and low utility. Particularly, the logically symmetric Laplace distribution is used in the temporal perturbation to keep the uniform distribution of perturbed measurements. With the Laplace noises distribution, the time information in the real-time measurements is differentially private. Besides, perturbed time-series measurements will cause the distortion of individual measurements the system server received, thus protecting measurement privacy with filter-resistance. While, the system server is designed to process and extract crowd statistics from the received measurements via comparing the perturbed time stamps with current time. With the probability characteristics of Laplace distribution, the aggregation in a time slot can be estimated in a real-time way from the received measurements, which form a sample space of the large numbers of measurements. Besides real-time aggregation, off-line accumulation can be precisely computed from the temporally perturbed measurements on the server side. Therefore, our scheme can also achieve enough data utility in real-time monitoring systems.

- **Twofold utility-privacy tradeoffs:** Instead of purely delaying the sensor measurements, combined with temporally symmetric perturbation approach (i.e., delaying or shifting to an earlier time) and effective extracting processes on the system server, our scheme can support both the *real-time aggregation* and *off-line accumulation for individuals* and achieve two different utility-privacy tradeoffs for real-time aggregation and off-line accumulation. On one hand, with temporal perturbation and real-time extraction from limited measurements received, the real-time disclosure is significantly reduced thus our scheme achieves better privacy; on the other hand, because the value of measurements is generally kept with temporal perturbation, the off-line accumulation can utilize all perturbed measurements and obtain better data utility supporting.

- **Extensive analysis and evaluation:** We conducted a combination of both theoretical analysis and simulation experiments on a case study of the smart metering system, to evaluate the effectiveness of our proposed scheme in comparison with temporal perturbation based scheme with exponential delay algorithm (EDA) [17], the norm data perturbation based scheme (NDP) [10], and the baseline Laplace data perturbation based scheme (LDP) [12], which apply the standard Laplace perturbation at each time stamp, in terms of both temporal privacy and measurement privacy, filter-resistance as well as utility-privacy tradeoffs. Our data shows that our scheme achieves better utility-privacy tradeoffs than existing schemes. Especially, we can achieve superior utility-privacy tradeoffs in off-line accumulation and better filter-resistance than the data perturbation based schemes.

The rest of the paper is organized as follows: Section 2 reviews the related work on perturbation-based privacy preserving and temporal privacy; Section 3 introduces the system model and the problem definitions. In section 4, we provide the overview of our proposed temporal perturbation scheme. Section 5 and 6 present the analysis on privacy and data utility respectively. Extensive simulations are presented in Section 7. Finally, Section 8 concludes this paper.

2. Related work

To mitigate the privacy issues in various fields related to WSNs such as wired and wireless networks, a number of privacy-preserving schemes have been extensively studied [24][10][25][26][21][27][5][28][27][20][29][18][17][30][6]. According to the different perspectives, these schemes can be categorized into: data-oriented and context-oriented [20].

Data-oriented privacy: Data-oriented privacy-preserving focuses on the transformation of sensor measurements. One typical technique is to add additive or multiplicative random noises to achieve an intuitive tradeoff between privacy and utility [10][24][25][26]. For example, Lin *et al.* [10] proposed to introduce noises in order that an aggregator can only obtain accurate aggregation information without access to individual measurements. Another technique is to add generally Laplace noises to guarantee differential privacy [12][5][27][28], which is a provable privacy metric proposed by Dwork [21]. For example, Fan *et al.* [5] proposed an adaptive sharing algorithm with differential privacy for the real-time aggregation system. However, data-oriented schemes are vulnerable to noise-filtering attacks [31][32] due to temporal correlations, not generic for various applications, lose utility in terms of single measurement, and have less considerations for context privacy.

Context-oriented privacy: Context-oriented privacy-preserving aims at protecting the context information of sensor data, i.e., the spatio-temporal information. Here, we mainly focus on temporal privacy, which concerns the time and temporal correlations of the sensitive data. Time information could analyse privacy from different sources and lead to the linkage attack (or reference attack) [33][34]. For example, Zhou *et al.* [35] proposed that the time stamps in the mobile phone traffic can infer the tweeting events via linking with the time stamps of users' tweets. Hence, it is critical to break the temporal correlations through temporal perturbation, which has following advantages than data perturbation: 1) flexible for various time-series based applications; 2) keeping better data utility in terms of measurements; 3)more filter-resistance to filtering attacks. One application considers temporal privacy is the Location Based Services (LBSs) [18][20][29]. For example, Assam *et al.* [29] made the trajectory anonymity with the delay time of the absolute value of Gaussian noise, which delays all the activities. Besides the study in LBSs, temporal perturbation has also been used general WSNs [6][17][30]. To obfuscate the events detected by the sensors, Kamat *et al.* [17] proposed to locally buffer the measurements in the intermediate nodes for a exponential delay in the delay-tolerant WSNs. However, the exponential delay would ruin the timeliness of dynamic trends. Guerreiro *et al.* [6] proposed the probabilistic sampling mechanism for WSNs, which can hide traffic patterns and user behaviors in data collection. However, this paper mainly focused on single sensor meter and did not design the server side to extract crowd dynamics with high data utility.

Different from the existing schemes, in this paper we developed a temporal perturbation based privacy-preserving scheme for real-time monitoring systems, which consider both temporal privacy and data privacy with high timeliness and utility-privacy tradeoff. Our proposed scheme can support both the real-time aggregation and off-line accumulation for individuals with sufficient privacy protection.

3. System model and problem statement

3.1. System model

A real-time monitoring system consists of three entities: sensor meters, the system server and the third-party users, as shown in Fig. 1. On the sensor meter side, the measuring component measures raw measurement periodically (i.e. fixed time slot). Then the raw measurement is processed to generate sensor readings at the processing component. Last, the delivery component buffers the measurement reports in the queue according to the processing component and sends the measurement reports to the system server via communication networks. On the system server side, the receiving and storage component gathers and stores the measurements to the database. Then the third-party users access to the database in the system server, analyse the aggregation results of sensor measurements, and publish the analysis results to relevant users. Here, we assume that large number of users (sensor meters) participate in the real-time monitoring system.

3.2. Threaten model

In this paper, we mainly concentrate on the privacy threatens, especially the behavioural privacy, which concerns that user' behaviours could be inferred from related features reflected in the time-series measurements [1][36]. To specific our

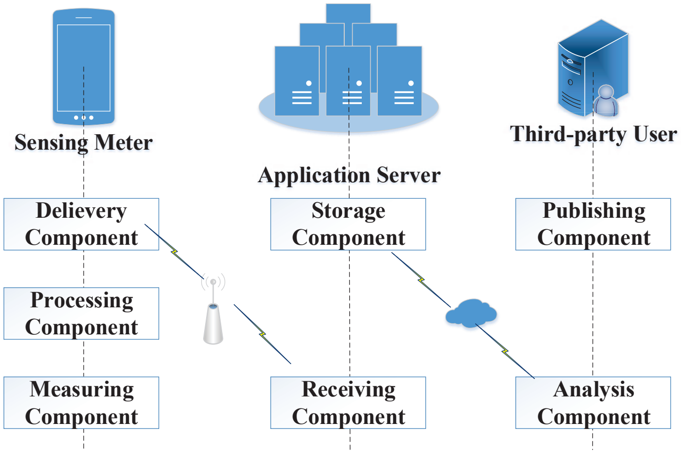

Figure 1: The components of the real-time monitoring system

adversaries and threatens, we have to mention the differences between privacy and (semantic) security. In the privacy domain, the data analyst, who receives the query information from the data curator, is not only a legitimate receiver but also a snooping adversary curious about the individual privacy. Hence, we cannot directly realize privacy protection via direct end-to-end encryption like anti-eavesdropping in security domain [9]. Because, either no information could be learnt without decryption, or total exposure with correct decryption, which does not satisfy with the object of a privacy-preserving query mechanism.

To focus on behavioural privacy issue, we assume that all components on the participants' side are trustworthy and the end to end communication is secured via existing authentication and encryption schemes. Also, instead of traditional adversaries in the semantic security domain (i.e., the eavesdroppers), in this paper, we consider an *honest-but-curious* adversary. An honest-but-curious adversary generally follows the processing correctly and does not provide false information on purpose. However, he is curious to probe the detailed privacy of the participants. For example, A typical honest-but-curious adversary can be the insider operator who secretly sells out participants' profiles. He is interested in the individuals' time-series measurements, and attempts to infer the detailed information about participants' behaviors and activities. Lastly, an honest-but-curious adversary may know all the algorithms and public parameters(i.e., the noise distribution) in our schemes.

3.3. Problem statement

The measurement of i^{th} participant's sensor meter M_i collected in the j^{th} time slot (one metering cycle, i.e., one minute) can be denoted as E^i_j. The collecting time of the measurement is T_j and can be recorded by its time-stamp

$w(T_j)$. Then, a vector, (E_1^i,E_2^i,...,E_j^i), can be used to define the trajectory or dynamics of the participant i. Besides the normal data privacy of sensor measurements at certain time points, more information can be mined from the time information and temporal correlations in highly ordered time-series measurements. Hence, we generalize and divide privacy issues in the real-time monitoring systems in two categories: *Temporal privacy* and *Measurement privacy.*

- ***Temporal privacy***: Temporal privacy means that an adversary cannot infer the correct time points T_j(time-stamp $w(T_j)$) or relative time order j of sensor measurements from the measurement reports.
- ***Measurement privacy***: Measurement privacy refers whether an adversary can obtain the original sensor measurements E_j^i or its precise estimation from the measurement reports.

Generally, data utility in the real-time monitoring systems is measured by two aggregation functions: *Real-time aggregation* and *Off-line accumulation* of sensor measurements for both timely estimations and individual summaries on the system server side. As remarked before, real-time aggregation can be used to monitor crowded dynamics, while off-line accumulation can provide individual awareness and bills.

- ***Real-time aggregation***: Real-time aggregation means the aggregation $\sum\limits_i E_j^i$ of real-time(i.e. the current time is T_j and the j^{th} time slot) sensor measurements E_j^i of all participants.
- ***Off-line accumulation***: Off-line accumulation for individual refers to the sum $\sum\limits_{j=1}^{T} E_j^i$ of sensor measurements E_j^i of each participants i in a continuous time period.

Intuitively, there is always contradiction between the individual privacy and data utility: high data utility will sacrifice the individual privacy and high privacy means little data fidelity. Thus, our problem is: 1) to find a general mechanism to balance the tradeoff between the privacy and data utility in the real-time monitoring systems; 2)quantify and verify the level of the privacy, data utility, and their tradeoff.

All notations used in this paper are summarized in Table 1.

4. Temporal perturbation based privacy-preserving scheme

4.1. Basic idea

The basic idea of our scheme is described below. In each time slot, once the measurements are collected, the time information of sensor measurements are logically perturbed to achieve differential privacy according to the Laplace distribution. Therefore, compared with its current time slot, the time slot of the sensor measurement after perturbation can be kept, delayed, or shifted to an

Table 1: Notation

M_i :	Meter ID of participant i.
T_j :	Reading time of the j^{th} time slot.
bT_j :	bT_j the time stamp of the beginning time of the j^{th} time slot, $[bT_j, bT_{j+1}]$ refers to the whole j^{th} time slot.
E^i_j :	Meter reading of participant i in the time slot j.
E'^i_j :	Temporally perturbed meter readings of participant i in the time slot j.
R_j :	The random delay time.
$\overline{X}^i_t$:	System server's temporal measurement of participant i in the current time slot.
$\hat{X}^i_t$:	System server's estimation measurement of participant i in the time slot t.
X^i_t :	System server's recorded measurement of participant i in the time slot t.
K_i :	Private key of participant i.
$H(\cdot)$:	Hash function stored in each sensor meter.
$w(\cdot)$:	Time stamp generating function for any time.
$J(w(T_j))$:	Mapping the time stamp $w(T_j)$ to the corresponding time slot j.

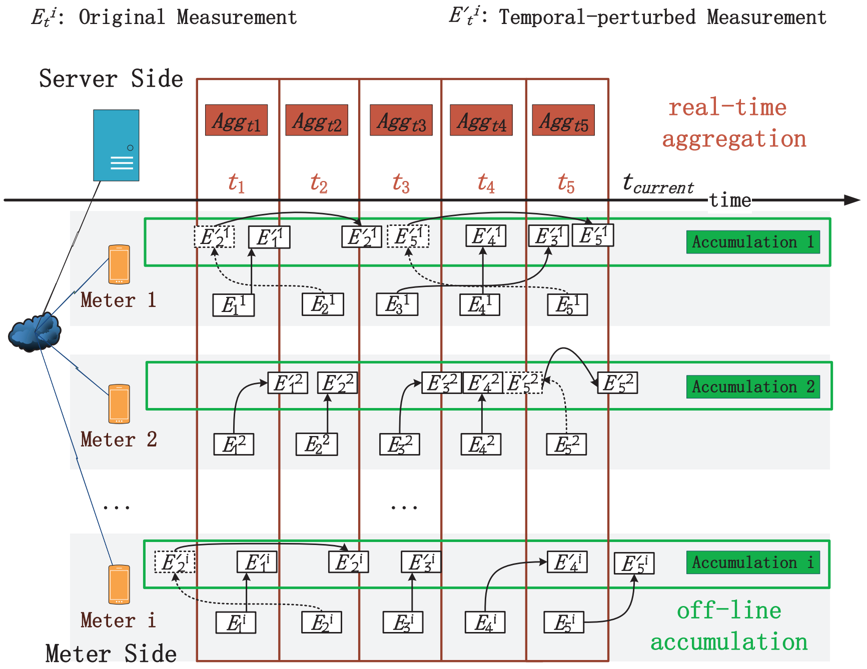

Figure 2: Architecture of the temporal perturbation based scheme[1]

earlier time, as shown in Fig. 2. **1)**If its time slot is kept in the current time slot

[1] The time of measurement is identified as their left bound in the figure, e.g., E'^1_1 belongs to the t_1 slot.

logically, then the sensor measurement report should be sent out immediately with modified time stamp, i.e., E'^{1}_{4}. It contributes to the most of the data fidelity. **2)**If its time slot is delayed logically, then the sensor measurement report should be sent out at the perturbed time with modified time stamp, i.e., E'^{1}_{3}. And the random delay can effectively protect the temporal privacy. **3)**If its time slot is shifted to an earlier time logically (e.g., the dotted box), then the sensor measurements should be modified with new time stamp and also randomly delayed to obfuscate its original time, i.e., $E_2'^{1}$. Thus, it is difficult for any receiver to infer the correct original time of the measurement report from its receiving time. Temporal privacy of sensor measurements are then protected. And distorted time-series measurements could also preserve the measurement privacy.

For the system server, we consider both the off-line and real-time cases. In the off-line case, because the perturbation is mainly on time information, the measurement readings keep the same. With sufficient long accumulation period, the off-line accumulation can be computed purely based on the sum of measurement readings, i.e., the green frame in Fig 2. In addition, the overall received measurements (off-line) in each time slot may consist of the original measurements collected in different time slots from a large number of sensors. Due to the temporal correlation of continuous time slots, all the perturbed measurements keep an approximation of original measurements. Besides, the number of received measurements also keeps nearly the same as it should be, due to the symmetry of temporal perturbation and the Central Limit Theorem [37] with large number of users. Hence, their off-line aggregation in each time slot also approximates to the original aggregation. However, in the real-time aggregation case, due to the nature that time does not go back, only the kept measurements in current slot and the delayed measurements received in current slot should be logically aggregated in real-time. Consequently, the real-time aggregation cannot be directly obtained as the off-line accumulation. For example, only E'^{2}_{2} can be real-time aggregated in the t_2 slot, both E'^{1}_{2} and E'^{i}_{2} will not be aggregated because they are logically shifted ahead. According to the characteristic of the Laplace distribution [38], we can know the distribution probability and estimate the original aggregation from the received measurements, which forms a sample space of the population. Therefore, the real-time aggregation (e.g., the coffee frame in the figure) can also be approximately estimated based on the Law of large numbers [39].

4.2. Sensor meter processing

The sensor meter processing mainly consists of the following three steps:

(1) ***Step 1: Raw data measuring:*** In our scheme, the measuring component of each sensor measures the raw data at each time slot to record participant-related readings. Therefore, at the j^{th} time slot, the sensor measures the raw data E^i_j, records the measuring time T_j, and generates measuring time stamp $w(T_j)$.

(2) ***Step 2: Time perturbation:*** For each measurement, the processing component first secretly generates a random seed $seed^i_j = H((M_i, T_j)_{K_i})$ and

generates a random number R_j according to the Laplace distribution $Laplace(0, b)$ with zero mean and the variance of $2b^2$. Then, the processing component modifies the measuring time and corresponding time-stamp as $T'_j = T_j + R_j$ and $w(T'_j) = w(T_j + R_j)$, respectively. Next, map the new time-stamp to the new time slot $j' = J(w(T'_j))$. Last, the measurement report r_j can be constructed by $r_j = (M_i, w(T'_j), j', E^i_j)$. We have to mention that we omit other auxiliary information(i.e. the MAC authentication code) here in the measurement report r_j.

(3) ***Step 3: Delay reporting:*** To perturb the time when the adversary observes the measurement report, the delivery component queues and delays the report packets as follows:

(i) If $w(R_j) \geq bT_j$, then buffer the measurement report in the queue, and send until the time of $S_j = T'_j = T_j + R_j$, and the sending time-stamp is $w(S_j)$.

(ii) If $w(R_j) < bT_j$, then the measurement report should be shifted to an earlier time logically, which breaches the nature that time doesn't go back. Hence, the sensor generates another random number D_j, which follows a exponential distribution $Exp(\lambda)$ with the variance of $1/\lambda^2$. And buffer the measurement report in the queue and send until time $S_j = T_j + D_j$, and the sending time-stamp is $w(S_j)$.

Repeat the above steps for each time slot and transmit all the measurements to the system server. We also mention that the whole end to end communication between the sensor meters and the system server should be kept secure with existing authentication and encryption techniques. Hence, the system server could normally receive and decode the proper measurement report r_j, which is supposed to be free from eavesdropping and tampering. Later, we will prove that the report r_j is privacy-preserving for the participants.

4.3. System server processing

At first, system server overall initializes the recorded measurement of each user at each time slot as 0, i.e., $\forall i, j, X^i_j = 0$. Then it repeats the following steps for each time slot:

(1) ***Step 1: Initialization:*** Initialize the real-time (current time slot t) aggregation $\overline{Agg_t}$ as 0. For each user i, initialize the temporal measurement of sensor reading as 0, i.e., $\forall i, \overline{X}^i_t = 0$.

(2) ***Step 2: Time extraction:*** Once the sensor measurement report r_j is received, system server first extracts the time slot j' in the measurement report r_j.

(3) ***Step 3: Real-time measurement update:*** If $j' = t$, then the measurement report r_j is regarded as the measurement at the current time slot t. So, the system server first updates the temporal measurement $\overline{X}^i_t$ as $\overline{X}^i_t = \overline{X}^i_t + E^i_j$, then updates the real-time aggregation $\overline{Agg_t} = \overline{Agg_t} + E^i_j$. If $j' \neq t$, then skip to step (4). Clearly, $t \neq T'_j$ actually means $t \geq T'_j$. That is to say this report is delayed and not reported timely for real-time aggregation. Therefore, the report is only need to be recorded in the database for later off-line accumulation for individuals.

(4) ***Step 4: Recorded measurement update:*** Every time measurement report r_j received will be recorded in the database, and add the measurement value E_j^i to user's recorded measurement of sensor reading at the time slot j', i.e., $X_{j'}^i = X_{j'}^i + E_j^i$.

4.3.1. Real-time aggregation

Clearly, real-time aggregation $\overline{Agg_t}$ will be produced in each time slot. From the above process, we know that those early shifted measurement reports will not be used for aggregation $\overline{Agg_t}$. Apparently, those early shifted measurement reports have a proportion of about 50% in the total measurement reports. Actually, all reports r_j with $w(R_j) < bT_j$ belong to the early shifted reports in our scheme. We normalize one time slot as 1 in the curve of Laplace distribution $Laplace(0, b)$. Then, according to the CDF(cumulative distribution function), $F(x, b) = \frac{1}{2}(1 + sgn(x)(1 - e^{-\frac{x}{b}}))$ of the Laplace distribution $Laplace(0, b)$, the proportion of the early shifted reports rates $F(-0.5, b) = \frac{1}{2}e^{-\frac{1}{2b}}$. So, the proportion of the timely reports collected in aggregation $\overline{Agg_t}$ is $(1 - \frac{1}{2}e^{-\frac{1}{2b}})$. These timely reports form a sample space of the total real-time measurements. Hence, their sum $\overline{Agg_t}$ also has a proportion of $1 - \frac{1}{2}e^{-\frac{1}{2b}}$ in the total aggregation Agg_t. Consequently, the real aggregation value Agg_t could be estimated as

$$Agg_t = \frac{2}{2 - e^{-\frac{1}{2b}}} \cdot \overline{Agg_t}. \tag{1}$$

4.3.2. Off-line accumulation for individuals

When system server needs to compute the accumulation for participants, according to the accumulation period, the corresponding reported measurements stored in the database at the system server can be retrieved to compute the individual accumulation by $Acc_i = \sum_t X_t^i$ in each accumulation cycle, which is composed of many continuous time slots. Because the perturbation is on time information of measurement reports, the total accumulation is less changed in a long period and could be computed more or less accurately as its original value $\sum_t E_t^i$.

In addition, because all the measurements value E_j^i are kept still, then other aggregation functions independent of the time order, such as Min/Max, $Median$, $Variance$, and $Histogram$ could also be computed directly.

4.4. Discussion

4.4.1. Remark 1

Our scheme is easy to have event-level privacy for time-series data, but hard to achieve user-level privacy [40]. Especially in the scenarios with steady measurements, the measurements have less changes in a period of time. Hence, the minor temporal perturbation could still disclose the user-level privacy in the measurement stream, because measurements in a continuous time period is difficult to be masked. Therefore, we propose an enhanced scheme with considering the data dividing strategy. The basic idea is to randomly divide the sensor readings into multiple samples and then perturb the time information on each sample. That is to add one more step of data dividing between the step 1 and step 2 of sensor meter processing. *Data dividing:* The processing component divides the raw measurement E_j^i into several samples randomly. For simplicity, we divide the raw measurement into two random samples $E1_j^i$ and $E2_j^i$ according to the uniform distribution and $E_j^i{=}E1_j^i{+}E2_j^i$. Time perturbation on these samples will cause the random combination in different time slots. Hence, the measurement privacy in each time slot is protected. This step is optional and tailored by the participants. To make it adaptive to any scenarios, we can set up with a deviation recorder that calculates the deviation of sensor readings in real-time. Once the deviation of current measurement series is lower than a configured threshold, the data dividing process should be triggered on.

4.4.2. Remark 2

In off-line accumulation, all measurements, including the logically early-shifted measurements, may need to be accumulated. However, time perturbation may cause some measurement"crosses the bounda" and lose utility in off-line accumulation, especially at the beginning time slot and the ending time slot of accumulation period. We propose two different strategies to overcome this issue: *a)Head-cutting:* the system server accumulates the recorded measurements according to the accumulation period and neglects the measurements "out of the boundary". This strategy will lose accumulation accuracy with short accumulation period. With longer accumulation period, the accumulation results could be more accurate. *b)Ring-moving:* imagine that the accumulation period is a ring, then with the MOD operation, the "out" measurements can be distributed along the ring of accumulation period. In this strategy, individual accumulation is lossless. However, with long accumulation period, the differences between the recorded measurements and the original measurements become large and lose utility. Hence, it is applicable to the short accumulation period.

4.4.3. Remark 3

Our scheme focuses on perturbation and would lose somewhat data fidelity on both the real-time aggregation and off-line accumulation. Due to our temporal perturbation, the time-series of aggregation results will have noises and fluctuate around the true value. Hence, we expect to reduce these noises and enhance the accuracy of these aggregation results. As a double-edged sword, filtering techniques can also effectively remove the white noises in time-series aggregations [5]. Hence, we can take the filtering algorithm to remove the noises in the aggregation results and enhance the accuracy and fidelity. Due to the large scale, aggregation results of participants' measurements are quite steady, which satisfies the premise for applying filtering techniques.

5. Privacy analysis

In this part, we will analyse both the temporal privacy and measurement privacy. Before the analysis, we first introduce the definition of **Expected Time Delay (ETD)**. ETD means the absolute expectation of the time delay in our perturbation scheme. According to the characteristics of Laplace distribution $Laplace(0, b)$, ETD equals to the parameter b, and shows the absolute time offset of the time perturbation.

5.1. Temporal privacy

We will analyse the temporal privacy achievement from three aspects. Firstly, we show the differential privacy in the temporal perturbation with Laplace noises. Secondly, we represent the temporal privacy in single measurement. Thirdly, we will show that the time orders in the measurement stream are broken by deriding the relationship of the mutual information in the perturbation processes.

5.1.1. Differential privacy

Based on our scheme design and definition of differential privacy [21], the time state of the measurement in our scheme is $1/b$-differentially private. According to the parallel composition property of differential privacy [41], the whole time series is $1/b$-differentially private. The detailed analysis can be referred in Appendix A.

5.1.2. Temporal privacy in single packet

After the intuitive analysis on indistinguishability, we need to specify the metrics of temporal privacy in the single measurement report. So, we define the **perturbation probability** to measure the probability that the time slot of the measurement report is distorted.

Definition 1. Perturbation probability (PP)*: Denote r_j^1,r_j^2,...,r_j^n as n dividing measurement samples of the original measurement E_j^i at the time slot j. And their corresponding time slots is denoted as ${j_1}'$,${j_2}'$,...,${j_n}'$, respectively. Then the perturbation probability PP_j of the time slot j can be denoted as,*

$$PP_j = p\left(({j_1}' \neq j) \cup ({j_2}' \neq j) \cup ... \cup ({j_n}' \neq j)\right). \tag{2}$$

So, according to the characteristic of the distribution $Laplace(0, b)$, the perturbation probability PP can be denoted as,

$$\begin{aligned} PP_j &= 1 - p\left(({j_1}' = j) \cap ({j_2}' = j) \cap ... \cap ({j_n}' = j)\right) \\ &= 1 - p\left((-\frac{1}{2} \leq R_j^1 \leq \frac{1}{2}) \cap (-\frac{1}{2} \leq R_j^2 \leq \frac{1}{2}) \cap ... \cap (-\frac{1}{2} \leq R_j^n \leq \frac{1}{2})\right) \\ &= 1 - (1 - e^{-\frac{1}{2b}})^n, \qquad (n = 1, 2, ...) \end{aligned} \tag{3}$$

where R_j^1, R_j^2,... are the random time delay in the report r_j^1, r_j^2,...,r_j^n, respectively.

From the above equation, we can see *the perturbation probability increases with the ETD (b) and number of samples (n)*. When b increases, the random time delay fluctuates dramatically. Besides, when n increases, there are more chances the reports in one slots are perturbed.

5.1.3. Temporal privacy in multi-packet stream

In this part, we will analyse the temporal information adversaries can obtain from measurement stream.

1) The sensor generates a stream of measurement packets $r_1, r_2, ..., r_j, ...$ with the corresponding time$(T_1, T_2, ..., T_j, ...)$. 2) The original time will be perturbed by a delay vector $(R_1,\ R_2,\ ...,\ R_j\ , ...)$, where each R_n is independently drawn from $Laplace(0, b)$. So, the time order after perturbation is $(T_1 + R_1, T_2 + R_2, ..., T_j + R_j, ...)$. We denote the perturbed time as $(Q_1, Q_2, ..., Q_j, ...)$, where $Q_j = T_j + R_j$. 3) Recall that R_n may be negative and there will be a second delay to propagate the measurements as in the step (3) of the sensor meter processing. The second delay D_j is non-negative and randomly chosen from the exponential distribution $Exp(\lambda)$ to replace the negative R_j. Hence, the new time is $(T_1 + \hat{R}_1, T_2 + \hat{R}_2, ..., T_j + \hat{R}_j, ...)$, where $\hat{R}_j = R_j$ when $R_j \geq 0$, or $\hat{R}_j = D_j$ when $R_j < 0$. It can be denoted as $(S_1, S_2, ..., S_j, ...)$. 4) After sorting, the system server will observe a new stream with the time $(\hat{S}_1, \hat{S}_2, ..., \hat{S}_j, ...)$, where $\hat{S}_j$ is the sorted series of S_j and $w(\hat{S}_1) \leq w(\hat{S}_2) \leq ...w(\hat{S}_j) \leq ....$ Consequently, the adversary's object is to infer the original time-series T_j from the final time-series $\hat{S}_j$. Denote $I(T, \hat{S})$ as the amount of information obtained by the adversary after observing the process $\{\hat{S}_j\}$. Then our privacy-preserving object is to make $I(T, \hat{S})$ as small as possible.

We briefly conclude here that *we need to tune λ small or b large to make $I(T, \hat{S})$, the adversary's information gain, small* and refer readers to Appendix B for the detailed analysis. Actually, tuning λ small or b large can reduce the information the adversary learns about the original measurement stream.

5.2. Measurement privacy

Besides the temporal information is privately kept, temporal perturbation can also lead to the perturbation on measurements. Then, we will represent the measurement privacy in terms of both the real-time data and recorded data.

To evaluate the measurement privacy, we apply following metrics to quantify the distortion of measurements: 1) **Distortion Standard Deviation (DSD)** [26], which is defined as the standard deviation between the original measurements and the distorted measurements; 2) **Cosine Similarity (Cosim[2])** [42], which is defined as the cosine distance of two vectors; 3) **Information Entropy**. According to the definition, higher distortion standard

[2] Distortion standard deviation can quantify the overall distance of two measurements stream, while cosine similarity is more effective to measure the distance in high-dimensional space. Here, the measurement stream can be regarded as a vector.

deviation means better privacy, while smaller cosine similarity leads to better privacy. Besides, less entropy also means less information revealing and better privacy.

5.2.1. Real-time data

Due to the characteristics of the Laplace distribution $Laplace(0, b)$, the cumulative distribution function of the random time delay R_j is $F(x, b) = \frac{1}{2}(1 + sgn(x)(1 - e^{-\frac{x}{b}}))$. So, from the view of the system server, the temporal measurement $\overline{X}_t^i$ of user i at the current time slot t should be

$$\overline{X}_t^i = (1 - e^{-\frac{1}{2b}})E_t^i + \frac{1}{2}\sum_k (e^{-\frac{2k-1}{2b}} - e^{-\frac{2k+1}{2b}})E_{t-k}^i, \tag{4}$$

where, E_t^i denotes the sensor reading at t^{th} time slot. Readers can refer to Appendix C for the detailed analysis. Hence, the temporal measurement $\overline{X}_t^i$ is the exponential smoothing of the original measurement series E_t^i. It aggregates $(1 - e^{-\frac{1}{2b}})$ percent current measurements E_t^i and predicts the rest with the history measurements E_{t-k}^i. The EWMA(exponentially weighted moving average) measurement not only preserves the measurement privacy but also keeps a good estimation on time-series measurement.

The real-time reported measurements have a proportion of $1 - \frac{1}{2}e^{-\frac{1}{2b}}$, then we can estimate the real-time measurement $\hat{X}_t^i$ at the t^{th} by

$$\hat{X}_t^i = \overline{X}_t^i / (1 - \frac{1}{2}e^{-\frac{1}{2b}}) = \frac{2 - 2e^{-\frac{1}{2b}}}{2 - e^{-\frac{1}{2b}}}E_t^i + \sum_k \frac{e^{-\frac{2k-1}{2b}} - e^{-\frac{2k+1}{2b}}}{2 - e^{-\frac{1}{2b}}}E_{t-k}^i. \tag{5}$$

For simplicity, we denote q_{-k} as the fraction of each measurement E_{t-k}^i in the Equation (5), that is to say,

$$q_0 = \frac{2 - 2e^{-\frac{1}{2b}}}{2 - e^{-\frac{1}{2b}}}, \tag{6}$$

$$q_{-k} = \frac{e^{-\frac{2k-1}{2b}} - e^{-\frac{2k+1}{2b}}}{2 - e^{-\frac{1}{2b}}}. \tag{7}$$

clearly, $\sum\limits_{k=0}^{\infty} q_{-k} = 1$ and q_{-k} decreases with b.

According to the Equation (5), the raw measurement E_t^i weights q_0 in the estimated measurement $\hat{X}_t^i$. Hence, when b increases, then q_0 decreases, and the difference between $\hat{X}_t^i$ and E_t^i becomes large, which also means better measurement privacy. About the information entropy, we can conclude that the information entropy in our perturbed measurements is smaller than that in the original measurements, and decrease with the increase of b. *This means the perturbed measurements reveal less information in the original measurements.* The detailed analysis can be referred in Appendix D.

5.2.2. Recorded data

Besides the real-time estimated measurement $\hat{X}_t^i$, the system server will also maintain a series of recorded data X_t^i, which is constantly updated with both the early shifted and delayed measurements. With time goes on and the update of received measurements, recorded measurement X_t^i becomes stable though it is not real-time. Usually, the real-time data is used to capture the trends or dynamics of participants immediately when they are received, while the recorded measurement can be used for more reliable statistics and analysis later. From the symmetric characteristics of the Laplace distribution $Laplace(0, b)$ and the Equation (4), we can deduce that the recorded measurement X_t^i should be

$$X_t^i = (1 - e^{-\frac{1}{2b}})E_t^i + \sum_k \frac{1}{2}(e^{-\frac{2k-1}{2b}} - e^{-\frac{2k+1}{2b}})(E_{t-k}^i + E_{t+k}^i). \tag{8}$$

Similarly, X_t^i is also the exponential smoothing of all measurements in all time slots. Hence, there are similar conclusions for the recorded measurements.

5.3. Filter-resistance

Many filtering techniques like the Kalman Filter [43] or PCA (Principal component analysis) [44] can produce relatively precise estimations via noised observations of time-series and lead to the exposure of individual privacy [32]. The data based perturbation schemes, which add the random noises, may suffer a lot from the noise filtering algorithms due to temporal correlations. Differently, temporally perturbed measurements could break the time correlations with shuffled time orders, thus forming a more filter-resistance input for filtering algorithms. Obviously, greater temporal distortion of measurement orders, means stronger filter-resistance.

Then, to quantify the distortion level of the whole measurement stream as well as the filter-resistance, we define the **shuffling probability** to measure the ratio that the relative time order of measurements are exchanged. Different form perturbation probability, shuffling probability measures the ratio that the time order of measurements are mutually exchanged and shuffled.

Definition 2. Shuffling probability (SP)*: Denote j_k and j_{k+1} as the neighbouring time slots mapping to the time order of two continuous original measurements $E_{j_k}^i$ and $E_{j_{k+1}}^i$. ${j'}_k^1$,${j'}_k^2$,...,${j'}_k^n$ and ${j'}_{k+1}^1$,${j'}_{k+1}^2$,...,${j'}_{k+1}^n$ denote the corresponding time slots after the time perturbation of n dividing measurement shares, respectively. Then the shuffle probability SP can be denoted as,*

$$\begin{aligned} SP &= p\left(\exists {j'}_k^x, {j'}_{k+1}^y, {j'}_k^x \geq {j'}_{k+1}^y\right), x, y = 1, 2, ..., n. \\ &= p\left(\max({j'}_k^1, {j'}_k^2, ..., {j'}_k^n) \geq \min({j'}_{k+1}^1, {j'}_{k+1}^2, ..., {j'}_{k+1}^n)\right). \end{aligned} \tag{9}$$

Let $R_{j_k}^x$ and $R_{j_{k+1}}^y$ denote the random time delay generated for the x^{th} dividing reports of r_{j_k} and the y^{th} dividing reports for $r_{j_{k+1}}$. Then, ${j'}_k^x \geq {j'}_{k+1}^y$ actually means $R_{j_k}^x - R_{j_{k+1}}^y \geq 1$. We define the event A_{xy} as $R_{j_k}^x - R_{j_{k+1}}^y \geq$

1, where R_{j_k} and R_{j_k} are independent with each other. Then, according to the characteristic of the linear combination of Laplace random variables [45], $p(A_{xy}) = 1 - \frac{1}{2}e^{-\frac{1}{b}}$. Hence, we can further obtain that

$$SP = \sum_{x=1}^{n}\sum_{y=1}^{n} p(A_{xy}) = 1 - \prod_{x=1}^{n}\prod_{y=1}^{n} p(A_{xy}) = 1 - \left(1 - \frac{1}{2}e^{-\frac{1}{b}}\right)^{n^2}, \qquad (10)$$

where, n represents number of divided measurement shares. Also, the shuffling probability increases with n and b. The greater ETD (b) means strong perturbation, also the greater the n is, the more chances the reports are shuffled and the perturbed time-series measurement will be more filter-resistance.

6. Utility analysis

In this part ,we mainly analyse the data utility in terms of real-time aggregation and off-line accumulation on the system server side.

6.1. Real-time aggregation

As we noted before, according to the Equation (4), the system server should obtain the real-time aggregation $\overline{Agg_t}$ at the current t^{th} time slot as

$$\overline{Agg_t} = \sum_i \overline{X}_t^i = (1 - e^{-\frac{1}{2b}})\sum_i E_t^i + \sum_k \frac{1}{2}(e^{-\frac{2k-1}{2b}} - e^{-\frac{2k+1}{2b}})\sum_i E_{t-k}^i, \qquad (11)$$

and the real-time aggregation Agg_t could be estimated as follows,

$$\begin{aligned} Agg_t &= \frac{2}{2 - e^{-\frac{1}{2b}}}\overline{Agg_t} \qquad (12) \\ &= \frac{2 - 2e^{-\frac{1}{2b}}}{2 - e^{-\frac{1}{2b}}}\sum_i E_t^i + \sum_{k=0}^{+\infty}\sum_i \frac{e^{-\frac{2k-1}{2b}} - e^{-\frac{2k+1}{2b}}}{2 - e^{-\frac{1}{2b}}} E_{t-k}^i \\ &= q_0 \sum_i E_t^i + \sum_{k=0}^{+\infty}\sum_i q_{-k} E_{t-k}^i. \end{aligned}$$

Similarly, the both Equation (11) and (12) form a exponential smoothing predicting model which combines both the current aggregation $\sum\limits_i E_t^i$ and history aggregations $\sum\limits_i E_{t-k}^i$ to smoothly estimate the current aggregation. Intuitively, Table 2 lists the weights of different time slots in the Equation (12) vs. $ETD(b)$. We can see, the fractions of neighboring 5 slots weight more than 91% in the whole estimated aggregation Agg_t. Based on the correlations of measurements, we can conclude an assumption that aggregation in neighbouring slots are steady and similar. Hence, the estimated aggregation in Equation (12), which consists of several aggregations in continuous time slots, could effectively draw a reliable estimation of real-time aggregation.

Table 2: Weights of different time slots versus ETD

Expected time delay (b)	0.5	1.0	1.5	2.0
Current time slot(q_0)	0.7746	0.5647	0.4417	0.3623
Last 1 time slot(q_{-1})	0.1949	0.2751	0.2716	0.2509
Last 2 time slot(q_{-2})	0.0264	0.1013	0.1395	0.1522
Last 3 time slot(q_{-3})	0.0036	0.0372	0.0716	0.0923
Last 4 time slot(q_{-4})	0.0005	0.0137	0.0368	0.0560

6.2. Off-line accumulation for individuals

Individual accumulation Acc_i is computed off-line. For example, the system server could calculate the participants' daily accumulation at the next day. Therefore, different from the above analysis of the real-time aggregation in the Equation (12), we should also consider the early shifted measurements in the off-line accumulation. Hence, the accumulation of i^{th} user before the time slot t is as follows.

$$Acc_i = (1 - e^{-\frac{1}{2b}}) \sum_{n=1}^{t} X_n^i + \sum_{k} \sum_{k \leq n}^{t} \frac{1}{2} (e^{-\frac{2k-1}{2b}} - e^{-\frac{2k+1}{2b}}) (X_{n-k}^i + X_{n+k}^i). \quad (13)$$

Because the recorded measurement is only a combination of original measurements in perturbed time slots. So, when many recorded measurements of a single user in continuous time slots are accumulated together, the perturbations can be cancelled and compensated with each other in the whole period. Therefore, the individual accumulation can be highly accurate for a predefined time period consists of many time slots.

7. Performance evaluation

In this section, we validate the effectiveness of our proposed scheme, by using the simulator developed by Richardson *et al* [46] and the MATLAB simulation tool. The organization of this section is as follows: First, we will illustrate our evaluation methodology including the scenario, dataset and metrics. Next, we present an illustrative case study to visualize the performance of our schemes. Then, we demonstrate the performance on both privacy and data utility of our scheme, by comparing it with other existing schemes. Finally, we discuss the filter-resistance performance, utility-privacy tradeoffs, and overheads of our scheme.

7.1. Simulation setup

7.1.1. Methodology

We take a case study on the real-monitoring scenario of the smart metering system (or AMI, advanced metering infrastructure) in the smart gird [2]. In the smart metering system, system server (or MDMS, Meter Data Management System) periodically collects the measurements from the smart meters via different communication technologies, as shown in Fig. 3 [47]. The fine-grained measurements can be aggregated to monitor the real-time grid states and accumulated for customer billing. Particularly, we considered a scenario consists of 1000 users and their electricity usages last for 1440 minutes (i.e., one day). For realistic simulations, the energy consumptions of 1000 randomized users with different family sizes, dwelling appliances and occupancy models were simulated and collected based on the simulator developed by Richardson *et al.* [46].

Based on the simulation data produced by the simulator [46], we implemented our scheme in Matlab and simulated the basic operations of real-time system aggregations and off-line accumulations for individual between the smart meters and the system server. Particularly, the smart meter reports to the system server in each time slot, and we perturb the time information in the measurements, which were retrieved according to the simulation data set. Then the system server collects the reported measurements at each time slot and compute the real-time aggregations and off-line accumulations according to our scheme.

7.1.2. Comparison

To show the effectiveness of our scheme, we also summarized, simulated and compared with the coral algorithms of relevant and typical schemes existed:

1. **Exponential Delay Algorithm (EDA)**: The delivery of measurement is randomly delayed by an exponentially random number sampled from $Exp(\lambda)$ [17].
2. **Norm Data Perturbation (NDP)**: Random noises with 0 mean and variance of σ^2 of normal distribution are added to the measurements. Due to the central limit theorem, noises in the aggregation results will be cancelled and a relatively accurate aggregation will be obtained [10].
3. **Laplace Data Perturbation (LDP[3])**: Random noise sampled from the Laplace distribution $Laplace(0, b)$ was added to the measurements in each time slot to achieve the differential privacy [12].

7.1.3. Metrics and parameters

We evaluate the performance on the following perspectives with different metrics in our simulations:

- **Privacy:** Temporal privacy is evaluated with perturbation probability(PP) defined in Section 5.1.2, shuffling probability(SP) defined in Section 5.3, and approximate entropy(Apen)[4] [48]; Measurement privacy is evaluated with distortion standard deviation(DSD) [26], cosine similarity(Cosim) [42], and information entropy introduced in Section 5.2. Entropy is measured by the distribution of time-series as in [36]. In addition, differential privacy metric [21] is also considered.

[3] To gain an intuitive knowledge, the Laplace noises here was calibrated with the local sensitivity instead of global sensitivity, which requires the global knowledge and is hard to be captured in the real-time monitoring. With the Local sensitivity, the LDP here has the best utility accuracy but inadequate differential privacy. That is to say, the LDP here achieves the impossibly best utility-privacy tradeoff.

[4] Larger approximate entropy means less repeated measurements. And less repeated measurements in our scheme means correlation destruction and better privacy.

- **Utility:** Mean square error(MSE) of both real-time aggregation and off-line accumulation are used to evaluate the data utility.
- **Filter-resistance:** The filter-resistance is evaluated with the mean square error(MSE) of filtering error.
- **Overhead:** The overhead is evaluated by the number of buffered measurement packets.

In our simulation, we considered both the dividing and non-dividing cases as discussed in Section 4.4.1. Specifically, in the non-dividing case, $n = 1$ and in the dividing case, $n = 2$, which divides the measurement into two samples following the uniform distribution. In addition, in the off-line accumulation, we used the Head-cutting strategy discussed in Section 4.4.2 to accumulate the recorded measurements dropped in the accumulation period of 1440 time slots for simplicity. The Apen is measured according the algorithm shown in [48], the main parameters compared length, and filtering level is set as $m = 2$ and $r = 50$, respectively. Besides, the information entropy is analysed as [36], where the parameter of bin size is set as $bin = 50$. We note that the mean and standard deviation of simulation data set generated via [46] is 273.15 and 581.32.

7.2. An illustrative case study

To present the performance of our scheme in an intuitive way, we show illustrative examples of the time perturbation, measurement perturbation and data aggregation (ETD is 1.0) in Fig. 4, 5, and 6, respectively. As we can see in Fig. 4, perturbed time of measurements is randomly distributed nearby its original time. Hence, the real-time estimated measurements of the randomly chosen user are largely distorted due to temporal perturbation, as shown in Fig. 5. In addition, as represented in Fig. 6, the real-time estimated aggregation on the system server fits well with the original aggregation and shows the similar dynamics, which shows the sufficient data utility in terms of real-time aggregation after temporal perturbation.

7.3. Privacy

7.3.1. Temporal privacy

Fig. 7 and Fig. 8 show both the analytical and experimental results of the perturbation probability and shuffling probability versus the ETD (b), respectively. Both the perturbation probability and shuffling probability increase with ETD gradually. Around $b = 1.0$, perturbation probability is more than 50%. That means at least half of measurements are time perturbed. When $ETD = 1.0$, the shuffling probability is about 20%, which means at least 20% measurements are shuffled in the measurement stream. All these shows that our scheme keeps the temporal privacy well. In Fig. 9, we mainly show the approximate entropy versus the ETD. The approximate entropy in perturbed measurements is generally larger than that in original measurements. This means our scheme has less repeated measurement pairs and breaks correlations in the measurement stream. Besides, the dividing case (n=2) has less approximate entropy than the non dividing case (n=1), this is because the over perturbation cause the uniformly scattering of measurements and large repeated measurement pairs. In addition, with the increase of ETD, approximate entropy keeps stable or even drops, the reason is similar.

7.3.2. Measurement privacy

Fig. 10 shows the distortion standard deviation of the measurements versus the ETD. As shown, the distortion standard deviation increases gradually with ETD, which shows the measurement distortion and privacy. Generally, recorded data has larger distortion due to more data (including early shifted data) perturbed. Compared with the EDA scheme, our scheme has larger distortion standard deviation and better privacy because of the symmetric perturbation. Fig. 11 shows the results of the cosine similarity between the perturbed measurements and original measurements. As shown, our data shows less cosine similarity, which also means greater differences and higher privacy level. In Fig. 12, we present the information entropy in the perturbed measurements versus the ETD. The information entropy clearly decreases and is less than that in the original measurements. This means the perturbed measurements reveal less information and keep privacy well, which corresponds to our analysis. The EDA algorithm has nearly the same entropy as the original measurements, which is because the measurements are only temporally delayed have less distortion. For brevity and fairness, otherwise specified, the simulation measurements in the following comparison is the non-dividing case (n=1).

7.4. Data utility

Fig. 13 presents the mean± standard deviation of the real-time aggregation error versus ETD. The mean value fluctuates around 0%, which shows the aggregation error converges to zero. Though the standard deviation increases with the ETD, its value is still limited and much small and the accuracy is enough to acquire the dynamic changes for system server. Furthermore, as shown in the figure, Kalman filtering technique can effectively enhance the accuracy of time-series aggregation without individual privacy concern. Fig. 14 shows the mean± standard deviation of the off-line accumulation error versus ETD. The mean value still fluctuates around 0% and is much smaller than in Fig 13. The standard deviation value is much smaller. The mean values are negative because some measurements near the beginning and ending time slots of the accumulation period are perturbed out. Nevertheless, these "out" reports could be accumulated and compensated at other accumulation period, which could keep the total accumulation lossless. It has to mention that, off-line accumulations are computed on individual users instead of time, so filtering techniques are not applied.

7.5. Filter-resistance

Fig. 15, shows the relative filtering error of Kalman filtering algorithm on both the our scheme and the data-oriented perturbation (with Laplace noises).

For simplicity of comparison, we also normalize the series of data to the domain of time series ($0 \sim 1440$), and use normal distribution to approximate the Laplace distribution. As shown, the filtering error in our scheme is much larger than that in the data-oriented perturbation, which means our scheme is more filter-resistant. This is because filtering algorithm performs better on the sequential data series and produces precise estimation with observations over time and predictive noises. In our temporal perturbation, the observations are distorted in terms of time orders. Consequently, it is difficult for filtering algorithms to predict the accurate time-series measurements.

7.6. Twofold utility-privacy tradeoffs

Fig. 16 and 17 show the twofold utility-privacy tradeoffs for both real-time aggregation and off-line accumulation in terms of differential privacy. As we can see, errors in both our scheme and the LDP scheme decrease with the increase of privacy budget (the decrease of privacy protection), which reflects the essence of the tradeoffs between privacy and data utility. As shown, the errors in LDP with large privacy budget (≥ 0.4) are nearly the same as our scheme, because the difference is not obvious when temporal perturbation is small. However, with both greater perturbation, temporal perturbation causes less errors because the shuffled measurements become stable gradually. While, the noises in LDP still increase exponentially. We also show the enhanced performance with the Kalman filtering technique on the real-time aggregation. In Fig. 17, our off-line accumulation has overwhelmingly higher accuracy because temporal perturbation is nearly lossless on accumulation, as discussed before. In Fig. 18, 19, 20, and 21, we also show the twofold utility-privacy tradeoffs in terms of distortion standard deviation and cosine similarity. Particulary, we fix the same privacy level(i.e., DSD and Cosim) and compare the data utility level(i.e., MSE). With the same privacy level, both our real-time aggregation and off-line accumulation have lower errors. Especially, the accumulation error is much smaller because the perturbation is only on the time dimension. Overall, our scheme has twofold better tradeoffs between data utility and privacy than the existing perturbation schemes.

7.7. Buffer size

In our scheme, measurements are buffered to perturb delivering sequence. Hence, enough buffer space is needed for the sensor meters. For simplicity, we focus on the number of buffered measurements. Because the cycle of the time slot is fixed, longer ETD means more measurements are buffered to delay delivering. It is easy to know that the expected number of buffered measurements is the ratio of the ETD and the cycle of time slot. We simply statistic the max and average number of buffer measurements which increase with ETD. And the max number of buffered measurements is 6 when ETD is maximized as 2 in our simulation. Considered that early shifted packets are also buffered, the max buffered reports number will be no more than 12 because the ratio of early-shifted reports is nearly 50%.

8. Conclusion

In this paper, we propose a novel temporal perturbation based privacy-preserving scheme for real-time monitoring system, which can effectively protect participants' privacy. Through adopting the symmetric Laplace random noises to perturb the delivery time of measurement reports, the proposed scheme preserves both the temporal privacy and measurement privacy of the whole measurement stream, while releasing reliable aggregation dynamics and off-line accumulation for individuals. Our scheme achieves a high data utility for both real-time aggregation and off-line accumulation with a low overhead. Both theoretical analysis and simulation experiments validate that our scheme achieves much better performance in both the privacy and utility, as well as resilience to the filtering attacks on perturbation schemes. Our work demonstrate that temporal perturbation is an effective way for preserving privacy in real-time monitoring systems. Our future work will focus on using spatio-temporal correlation perturbation of multiple data measurement streams for privacy preserving in real-time networked sensing systems.

Appendix A. The analysis of differential privacy

Though the sensor readings can be decoded by adversaries, the right collecting time of sensor readings still can not be determined in our scheme. Suppose that the measurement report time slot is t, and $p(j|t)$ and $p(j+1|t)$ denote the probability that the original time slot is j or $j+1$ when report time slot is t. Then,

$$\frac{p(j|t)}{p(j+1|t)} = \frac{noise(j-t)}{noise(j+1-t)} = \frac{e^{-\frac{|j-t|}{b}}}{e^{-\frac{|j+1-t|}{b}}} \leq e^{\frac{1}{b}}, \tag{A.1}$$

where $noise(j-t)$ is the random time delay R_j generated with the Laplace distribution $Laplace(0,b)$. Hence, when $1/b$ is small, the ratio of $p(k|t)$ and $p(k+1|t)$ is close to 1 ($e^{\frac{1}{b}} \approx 1+1/b$, when $1/b$ is small). That means, when the reported time slot is t, the probability that its original time slot belongs to j is quite close to the probability that its original time slot belongs to $j+1$. Hence, they are $1/b$-differentially private according the definition in [21]. Intuitively, adversaries can not distinguish at which time slot the measurement is reported. The smaller $1/b$ is, the more difficult for the adversary to distinguish and infer the correct time of measurement report, and the better temporal privacy is. In addition, according to the parallel composition property of differential privacy [41], the whole time series is $1/b$-differentially private.

Appendix B. The analysis of adversary's information

Due to the data processing $T_j \to Q_j \to S_j \to \hat{S}_j$, we can know the relationship $I(T,\hat{S}) \leq I(T,S) \leq I(T,Q)$. Then,

$$\begin{aligned} I(T,\hat{S}) &\leq I(T,S) \\ &= h(S) - h(\hat{R}) \\ &\leq \sum_{t=1}^{j} (h(S_t) - h(\hat{R}_t) = \sum_{t=1}^{j} I(T_t, S_t). \end{aligned} \tag{B.1}$$

As mentioned before, the perturbation noise $\hat{R}_j$ is a random variable mixed the Laplacian distribution and the exponential distribution as follows,

$$\hat{R}_j = \begin{cases} R_j, R_j \sim Laplace(0,b) \qquad (R_j \geq 0) \\ D_j, D_j \sim Exp(\lambda) \qquad (R_j < 0), \end{cases} \tag{B.2}$$

where, we can see $\hat{R}_j$ is either drawn from the positive side of the Laplacian distribution $Laplace(0,b)$ or the exponential distribution $Exp(\lambda)$. So, $\hat{R}_j$ is basically an exponential random variable with the rate parameter of either λ or $1/b$. And according the analysis in [17], we have that

$$\begin{aligned} I(T_t;S_t) &= I(T_t;T_t+\hat{R}_t) \\ &\leq \max(\ln(1+j\lambda), \ln(1+t/b)). \end{aligned} \tag{B.3}$$

Using the above result, we can obtain that

$$I(T,S) \leq \sum_{t=1}^{j} (\max(\ln(1+t\lambda), \ln(1+t/b))). \tag{B.4}$$

So, to make

$$I(T,\hat{S}) \leq I(T,S) \leq \sum_{t=1}^{j} (\max(\ln(1+t\lambda), \ln(1+t/b))) \tag{B.5}$$

as small as possible, we need to tune λ small or b large.

Appendix C. The analysis of temporal measurement

Due to the characteristics of the Laplace distribution $Laplace(0,b)$, the CDF(cumulative distribution function) of the random time delay R_j is $F(x,b) = \frac{1}{2}(1+sgn(x)(1-e^{-\frac{x}{b}}))$. So, when a measurement report is received at the current time slot, the probability it belongs to the current time slot is

$$p_0 = F(0.5,b) - F(-0.5,b) = 1 - e^{-\frac{0.5}{b}} = 1 - e^{-\frac{1}{2b}}, \tag{C.1}$$

and the probability it originally belongs to the last 1 time slot is

$$p_{-1} = F(1.5, b) - F(0.5, b) = \frac{1}{2}(e^{-\frac{1}{2b}} - e^{-\frac{3}{2b}}), \tag{C.2}$$

and the probability it originally belongs to the last 2 time slot is

$$p_{-2} = F(2.5, b) - F(1.5, b) = \frac{1}{2}(e^{-\frac{3}{2b}} - e^{-\frac{5}{2b}}), \tag{C.3}$$

and so on, the probability it originally belongs to the last k time slot is

$$p_{-k} = F(k + 0.5, b) - F(k - 0.5, b) = \frac{1}{2}(e^{-\frac{2k-1}{2b}} - e^{-\frac{2k+1}{2b}}). \tag{C.4}$$

So, from the view of the system server, the temporal measurement $\overline{X}^i_t$ of user i at the current time slot t should be

$$\overline{X}^i_t = (1 - e^{-\frac{1}{2b}})E^i_t + \sum_k \frac{1}{2}(e^{-\frac{2k-1}{2b}} - e^{-\frac{2k+1}{2b}})E^i_{t-k}, \tag{C.5}$$

where, E^i_t denotes the sensor reading at t^{th} time slot.

Appendix D. The entropy analysis

Usually, the measurements E^i_j approximately follow a normal distribution $N(\mu, \omega)$. However, the different E^i_j of one participant i at different time slots are believed not totally independent with each other. Since the physical event information is modeled to have an exponential autocorrelation function [49], and the covariance function ranges between 0 and 1 [23], we can know the sensor measurements have correlations $0 \le \rho_{ij} < 1$. Suppose that the variance $Var(E^i_j) = \omega^2$, then $\hat{X}^i_t$ follows an approximate normal distribution $N(\mu, \omega')$ and its variance should be

$$\begin{aligned}
{\omega'}^2 &= q_0^2 Var(E^i_t) + \sum_k {q_{-k}}^2 Var(E_{t-k}) \\
&+ 2 \sum_{1 \le i < j} \sum_{i < j \le n} \rho_{ij} q_{-i} q_{-j} Var(E_{t-i}) Var(E_{t-j}) \\
&= \left(q_0^2 + \sum_k {q_{-k}}^2 + 2 \sum_{0 \le i < j} \sum_{i < j \le n} \rho_{ij} q_{-i} q_{-j} \right) \omega^2 \\
&< \left(\sum_{k=0}^{+\infty} q_{-k} \right)^2 \omega^2 \\
&= \omega^2.
\end{aligned} \tag{D.1}$$

Besides, with the increase of b, the distribution becomes more flat, the p_k becomes more close with each other and $Var(\hat{X}^i_j)$ decreases. In a word, ${\omega'}^2 =$

$Var(\hat{X}_j^i)$ decreases with b. According to the relationship between the entropy $Ent(E_j^i)$ and the variance ω^2 of the normal distribution [50] $N(\mu, \omega)$, $Ent(E_j^i) = ln(\omega\sqrt{2\pi e})$, the entropy of the perturbed measurements $Ent(\hat{X}_j^i) < Ent(E_j^i) = ln(\omega\sqrt{2\pi e})$ and also decreases with the increase of b. That means the perturbed measurements reveal less information on the original measurements.

[1] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick. A survey on privacy in mobile participatory sensing applications. *Journal of Systems and Software*, 84(11):1928–1946, 2011.

[2] X. Fang, S. Misra, G. Xue, and D. Yang. Smart grid-the new and improved power grid: A survey. *IEEE Communications Surveys and Tutorials*, (99):1–37, 2011.

[3] F. M. Cleveland. Cyber security issues for advanced metering infrasttructure (ami). In *IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–5, 2008.

[4] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell. A survey of mobile phone sensing. *IEEE Communications Magazine*, 48(9):140–150, 2010.

[5] L. Fan and L. Xiong. Adaptively sharing real-time aggregate with differential privacy. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 26(9):2094–2106, 2013.

[6] J. Guerreiro, E. C. Ngai, and C. Rohner. Privacy-aware probabilistic sampling for data collection in wireless sensor networks. In *7th International Wireless Communications and Mobile Computing Conference (IWCMC'11)*, pages 314–319, 2011.

[7] C. Y. Chow and M. F. Mokbel. Trajectory privacy in location-based services and data publication. *ACM SIGKDD Explorations Newsletter*, 13(1):19–29, 2011.

[8] S. Drenker and A. Kader. Nonintrusive monitoring of electric loads. *IEEE Computer Applications in Power*, 12(4):47–51, 1999.

[9] C. Dwork and A. Roth. The algorithmic foundations of differential privacy. *Theoretical Computer Science*, 9(3-4):211–407, 2013.

[10] H. Y. Lin, W. G. Tzeng, S. T. Shen, and B. S. Lin. A practical smart metering system supporting privacy preserving billing and load monitoring. In *Applied Cryptography and Network Security (ACNS'12)*, pages 544–560, 2012.

[11] H. Y. Lin, S. T. Shen, and B. P. Lin. A privacy preserving smart metering system supporting multiple time granularities. In *IEEE 6th International*

Conference on Software Security and Reliability Companion, pages 119–126, 2012.

[12] C. Dwork, F. McSherry, K. Nissim, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the 3rd conference on Theory of Cryptography (TCC'06)*, pages 265–284, 2006.

[13] L. Fan, L. Xiong, and V. Sunderam. Differentially private multi-dimensional time series release for traffic monitoring. In *Proceedings of the 27th international conference on Data and Applications Security and Privacy (DBSec)*, pages 33–48. 2013.

[14] F. G. Mármol, C. Sorge, O. Ugus, and G. M. Pérez. Do not snoop my habits: preserving privacy in the smart grid. *IEEE Communications Magazine*, 50(5):166–172, 2012.

[15] J. Shi, Y. Zhang, and Y. Liu. Prisense: privacy-preserving data aggregation in people-centric urban sensing systems. In *IEEE International Conference on Computer Communications (INFOCOM'10)*, pages 1–9, 2010.

[16] K. Kursawe, G. Danezis, and M. Kohlweiss. Privacy-friendly aggregation for the smart-grid. In *Privacy Enhancing Technologies(PET'11)*, pages 175–191, 2011.

[17] P. Kamat, W. Xu, W. Trappe, and Y. Zhang. Temporal privacy in wireless sensor networks: Theory and practice. *ACM Transactions on Sensor Networks (TOSN)*, 5(4):28, 2009.

[18] R. H. Hwang, Y. L. Hsueh, H. W. Chung, X. Lian, L. Chen, J. X. Yu, J. Um, H. Kim, Y. Choi, and J. Chang. A novel time-obfuscated algorithm for trajectory privacy protection. *IEEE Transactions on Services Computing*, 7(2):126–139, 2014.

[19] M. E. Andrs, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-indistinguishability: Differential privacy for location-based systems. In *Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security(CCS'13)*, pages 901–914, 2013.

[20] N. Li, N. Zhang, S. K. Das, and B. Thuraisingham. Privacy preservation in wireless sensor networks: A state-of-the-art survey. *Ad Hoc Networks*, 7(8):1501–1514, 2009.

[21] C. Dwork. Differential privacy. In *Automata, languages and programming, Springer*, pages 1–12. 2006.

[22] Julius Kusuma, Lance Doherty, and Kannan Ramchandran. Distributed compression for sensor networks. In *Proceedings of 2001 International Conference on Image Processing (ICIP'01)*, pages 82–85, 2001.

[23] M. C. Vuran, Ö. B. Akan, and I. F. Akyildiz. Spatio-temporal correlation: theory and applications for wireless sensor networks. *Computer Networks*, 45(3):245–259, 2004.

[24] R. Agrawal and R. Srikant. Privacy-preserving data mining. *ACM Sigmod Record*, 29(2):439–450, 2000.

[25] X. He, W.and Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher. Pda: Privacy-preserving data aggregation in wireless sensor networks. In *26th IEEE International Conference on Computer Communications (INFOCOM'07)*, pages 2045–2053, 2007.

[26] X. Ren, X. Yang, J. Lin, Q. Yang, and W. Yu. On scaling perturbation based privacy-preserving schemes in smart metering systems. In *22nd IEEE International Conference on Computer Communications and Networks (ICCCN'13)*, 2013.

[27] E. Shi, T. H. H. Chan, E. G. Rieffel, R. Chow, and D. Song. Privacy-preserving aggregation of time-series data. In *Network and Distributed System Security Symposium (NDSS'11)*, page 4, 2011.

[28] G. Acs and C. Castelluccia. I have a dream!(differentially private smart metering). In *Information Hiding (IH'11)*, pages 118–132, 2011.

[29] R. Assam and T. Seidl. Preserving privacy of moving objects via temporal clustering of spatio-temporal data streams. In *Proceedings of the 4th ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 9–16, 2011.

[30] A. RayChaudhuri, U. K. Chinthala, and A. Bhattacharya. Obfuscating temporal context of sensor data by coalescing at source. *ACM SIGMOBILE Mobile Computing and Communications Review*, 11(2):41–42, 2007.

[31] M. Sramka. A privacy attack that removes the majority of the noise from perturbed data. In *The 2010 International Joint Conference on Neural Networks (IJCNN'10)*, pages 1–8, 2010.

[32] K. Liu, C. Giannella, and H. Kargupta. A survey of attack techniques on privacy-preserving data perturbation methods. *Advances in Database Systems*, 34:359–381, 2008.

[33] A. Narayanan and V. Shmatikov. Robust de-anonymization of large sparse datasets. In *IEEE Security and Privacy*, pages 111–125, 2008.

[34] G. Navarro-Arribas and V. Torra. Information fusion in data privacy: A survey. *Information Fusion*, 13(4SI):235–244, 2012.

[35] X. Zhou, S. Demetriou, D. He, M. Naveed, X. Pan, X. Wang, C. Gunter, and K. Nahrstedt. Identity, location, disease and more: Inferring your secrets from android public resources. In *Proceedings of the 2013 ACM*

SIGSAC conference on Computer and communications security(CCS'13), pages 1017–1028, 2013.

[36] S. McLaughlin, P. McDaniel, and W. Aiello. Protecting consumer privacy from electric load monitoring. In *Proceedings of the 18th ACM conference on Computer and communications security (CCS'11)*, pages 87–98, 2011.

[37] Wikipedia. *Central Limit Theorem.* `http://en.wikipedia.org/wiki/Central_limit_theorem/`.

[38] Wikipedia. *Laplace Distribution.* `http://en.wikipedia.org/wiki/Laplace_distribution`.

[39] Wikipedia. *Law of large numbers.* `http://en.wikipedia.org/wiki/Law_of_large_numbers`.

[40] C. Dwork. Differential privacy in new settings. In *ACM Symposium on Discrete Algorithms (SODA'10)*, pages 174–183, 2010.

[41] F. 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*, pages 19–30, 2009.

[42] Wikipedia. *Cosine Similarity.* `http://en.wikipedia.org/wiki/Cosine_similarity`.

[43] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, Chapel Hill, NC, USA, 1995.

[44] R. Ganti, N. Pham, Y. Tsai, and T. Abdelzaher. Poolview: stream privacy for grassroots participatory sensing. In *Proceedings of the 6th ACM conference on Embedded network sensor systems (SenSys'08)*, pages 281–294, 2008.

[45] S. Nadarajah and S. Kotz. On the linear combination of laplace random variables. *Probability in the Engineering and Informational Sciences*, 19(04), 2005.

[46] I. Richardson, M. Thomson, D. Infield, and C. Clifford. Domestic electricity use: A high-resolution energy demand model. *Energy and Buildings*, 42(10):1878–1887, 2010.

[47] Toshiba. *Advanced Metering Infrastructure (AMI) for Smart Grids.* `http://www.toshiba-smartcommunity.com/en/blog/ami-system`.

[48] Wikipedia. *Approximate entropy.* `http://en.wikipedia.org/wiki/Approximate_entropy/`.

[49] L. Süber, Gordon. *Principles of mobile communication.* Springer, 2011.

[50] Wikipedia. *Normal distribution.* `http://en.wikipedia.org/wiki/Normal_distribution/`.

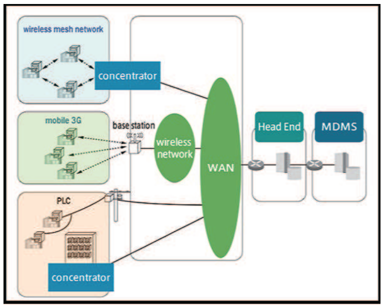

Figure 3: Smart metering system [47]

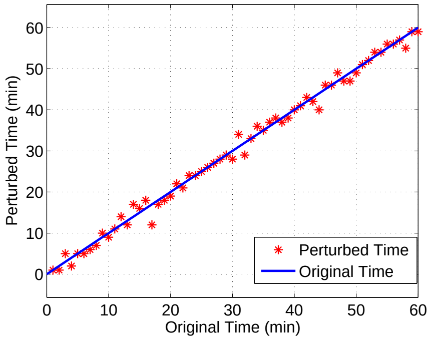

Figure 4: Time Perturbation $(b = 1.0)$

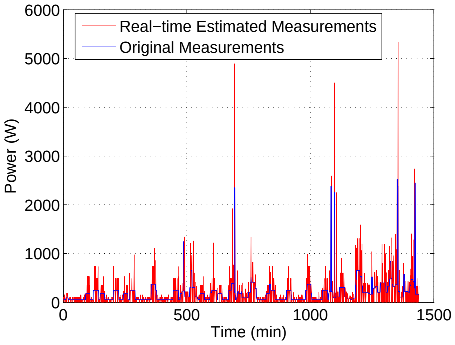

Figure 5: Comparison of the Load Profile $(b = 1.0)$

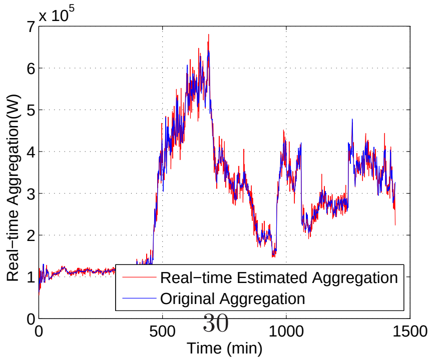

Figure 6: Comparison of Aggregation Results $(b = 1.0)$

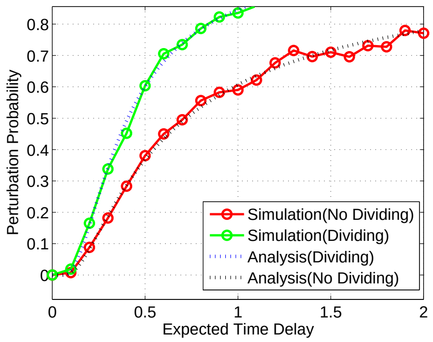

Figure 7: PP vs. ETD

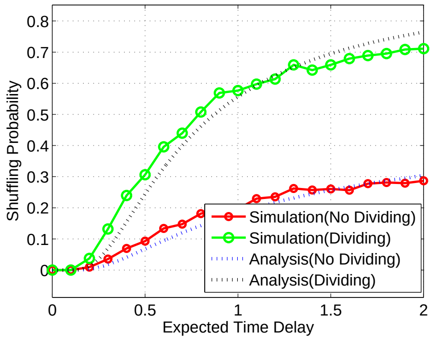

Figure 8: SP vs. ETD

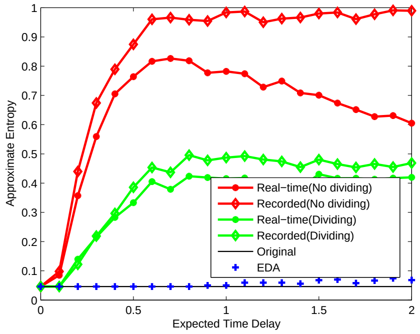

Figure 9: Apen vs. ETD

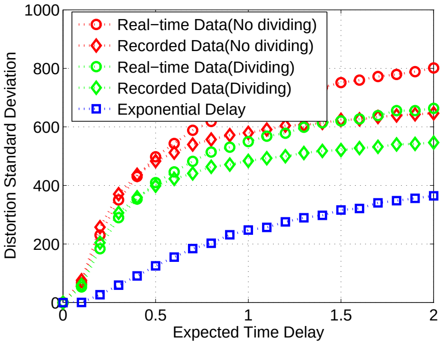

Figure 10: DSD vs. ETD

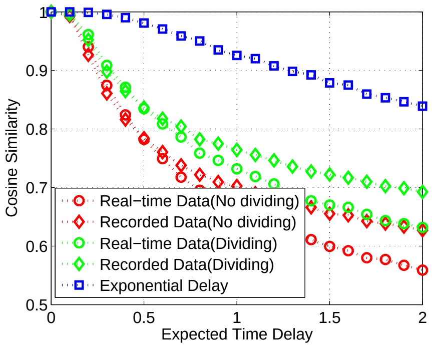

Figure 11: Cosim vs. ETD

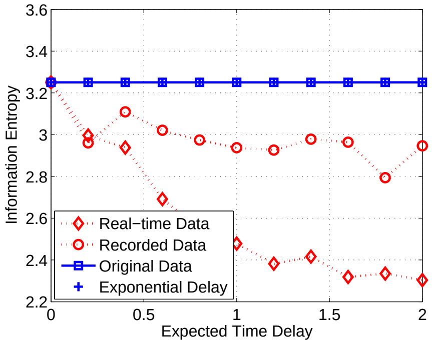

Figure 12: Entropy vs. ETD

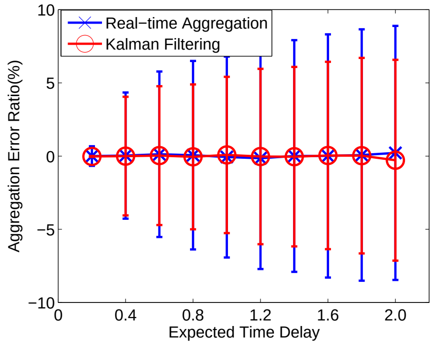

Figure 13: Aggregation Error vs. ETD

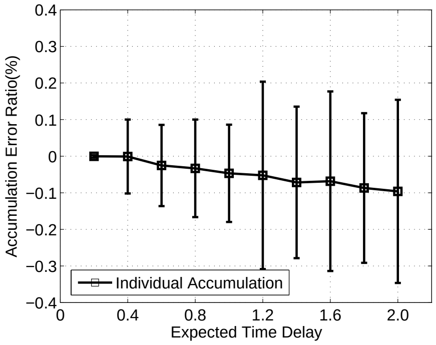

Figure 14: Accumulation Error vs. ETD

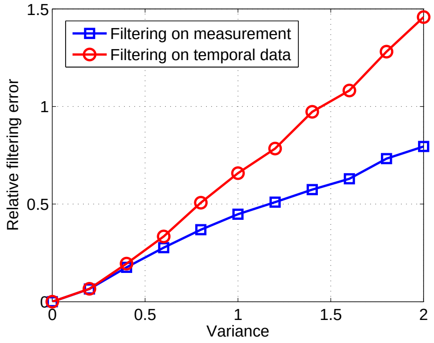

Figure 15: Comparison of Filter-resistance

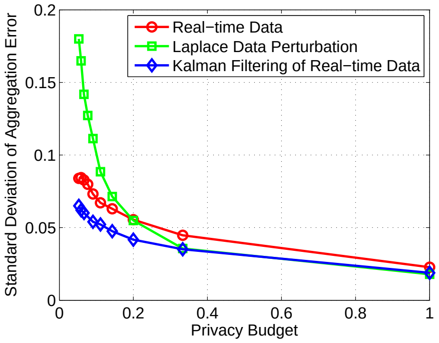

Figure 16: Aggregation Error vs. Differential Privacy Budget

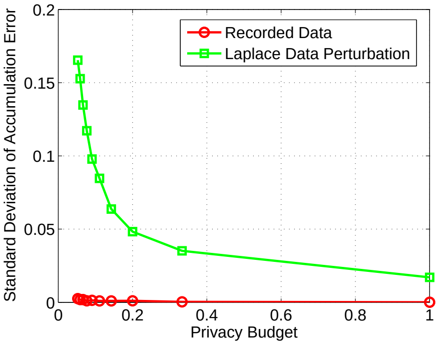

Figure 17: Accumulation Error vs. Differential Privacy Budget

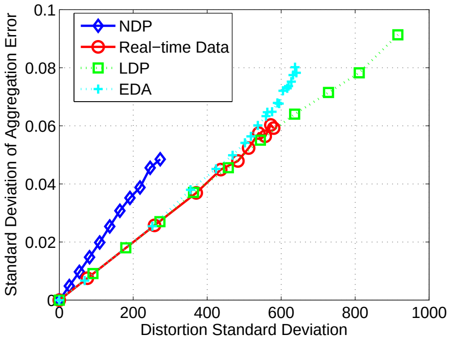

Figure 18: Aggregation Error vs. DSD

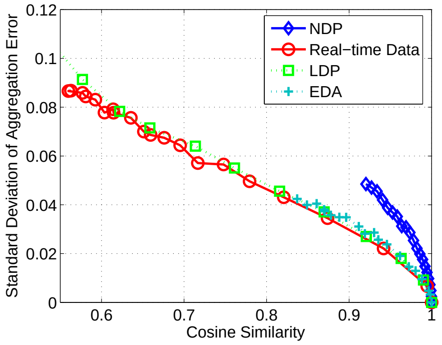

Figure 19: Aggregation Error vs. Cosim

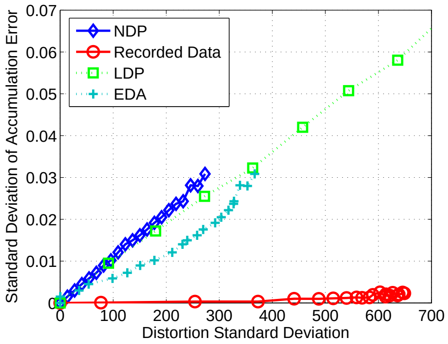

Figure 20: Accumulation Error vs. DSD

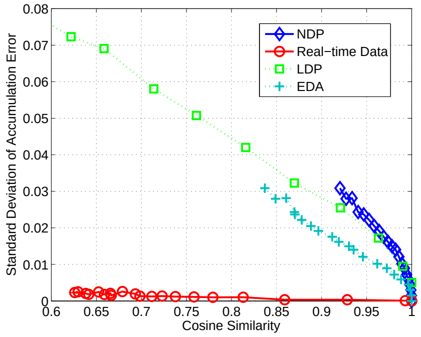

Figure 21: Accumulation Error vs. Cosim