A Privacy-Preserving Reputation System for Participatory Sensing

Kuan Lun Huang, Salil S. Kanhere
School of Computer Science Engineering
The University of New South Wales
Sydney, Australia
klh, salilk@cse.unsw.edu.au

Wen Hu
CSIRO ICT Center
CSIRO
Brisbane, Australia
Wen.Hu@csiro.au

***Abstract*—Participatory sensing is a revolutionary paradigm in which volunteers collect and share information from their local environment using mobile phones. The design of a successful participatory sensing application is met with two challenges - (1) user privacy and (2) data trustworthiness. Addressing these challenges concurrently is a non-trivial task since they result in conflicting system requirements. User privacy is often achieved by removing the links between successive user contributions while such links are essential in establishing trust. In this work, we present a way to transfer reputation values (which is a proxy for assessing trustworthiness) between anonymous contributions. We also propose a reputation anonymization scheme that prevents the inadvertent leakage of privacy due to the inherent relationship between reputation information. We conduct extensive simulations using real-world mobility traces and practical application. The results show that our solution reduces the probabilities of users being tracked via successive contributions by as much as 80%. Moreover, this improvement has no discernible impact on the normal operation of the application.**

I. Introduction

Advancements in mobile technologies in recent years have created a generation of sensor-rich mobile devices, which not only enable ubiquitous connectivity but are also equipped with PC-equivalent processing capability. These features have propelled the emergence of a new sensing paradigm that is now well-known as *participatory sensing* [1]. In participatory sensing, ordinary citizens collect data from their surrounding environment using their hand-held devices and upload them to an application server via existing communication infrastructure (e.g., 3G services or Wi-Fi access points). The application server then combines data from multiple participants, extracts the aggregate statistics, and uses the results to build a spatiotemporal view of the phenomenon of interest. Over the years, this revolutionary paradigm has been leveraged to design novel applications ranging from environmental monitoring [2], [3], [4], enhancing personal wellbeing [5] to identifying pricing dispersions in consumer goods [6].

The success of the above applications requires a high level of participation from users. To encourage participation, users must be assured that their confidential information, e.g., their identities or places visited, would not be disclosed as a result of their contributions. At the same time, the open nature of participatory sensing which allows anyone to contribute, exposes the application server to erroneous and malicious data. In other words, the design of a participatory sensing application is met with two challenges - (1) safeguarding user privacy and (2) ensuring data trustworthiness. Addressing these two challenges simultaneously is a non-trivial task because they result in conflicting system requirements. Specifically, user privacy is often provided via the use of pseudonyms and the obscuring of actual attribute (e.g., location) values, so that it becomes harder to establish links between multiple contributions from the same user. However, to build trust in a particular user, it is necessary to observe and rate multiple contributions made by that user over a period of time. In short, privacy requires connections between user contributions to disappear while data trustworthiness needs such connections to exist.

One approach to effectively evaluate the trustworthiness of data received from unknown users is to use a reputation system. We proposed one such system for participatory sensing applications in [8]. It functions by assigning a reputation value to each user as a measure of the trust placed on his contributed data. The main drawback of this system is that the reputation value is accumulative, i.e., it requires the system to know a user's historical behaviors to compute the current value. This requirement is in conflict with a system wherein users constantly change their pseudo-identities to preserve privacy, since reputation information is not transferred from one pseudonym to the next. In this work, we present a way to transfer user reputation information in a pseudonymous environment so that privacy and data trustworthiness can be simultaneously facilitated for participatory sensing applications. Our solution is based on a trusted third party server and does not require expensive cryptographic operations as necessary in prior work such as [9], nor does it incur as many communication overheads as [25]. Successful transfer of reputation information only solves part of the problem. As we show in Section II, even if reputation can be transferred from one anonymous contribution to the next, revealing user reputation allows an adversary to link consecutive user uploads. Successful linking of user contributions nullifies the protection from obscuring attribute values, e.g., the temporal and spatial anonymization introduced in [7], and leads to the de-anonymization of users. One approach to overcome this problem is to anonymize user reputations so that the transitions between reputation values are obfuscated. However, since the application server relies on user reputations to better estimate the output aggregate statistics, this process should not introduce significant errors when replacing the actual

reputation values. We propose an anonymization scheme based on the concept of k-anonymity [10] to prevent an adversary from de-anonymizing users while minimizing the impact on application outputs. Our specific contributions are as follows:

- We leverage a trusted server to transfer the reputation value for a user from one pseudonym to the next. The server maintains a list of mappings between the real user identity and all the associated pseudonyms. For each anonymous contribution made by a user, the server updates the corresponding reputation value and attaches it to the real identity. When a new upload is required, the server transfers the reputation value from the real identity to the next chosen pseudonym.
- We present a reputation anonymization scheme that eliminates the uniqueness in the transitions of user reputation. In essence, we ensure that a group of users share a common reputation value for each time interval. Our choice of algorithm minimizes the differences between actual and anonymized reputation values, so as to reduce the impact on the computation of application outputs.
- We conduct extensive simulations using real-world mobility traces and a practical participatory sensing application. Simulation results show that, as time elapses, our solution can reduce the linkability, i.e., the likelihood of linking contributions from the same user over time, by as much as 80%. In addition, the gain in linkability is not at the expense of the normal operation of the application.

The rest of this paper is organized as follows. Section II presents a motivating example which illustrates how reputation values are used to track pseudonymous users. We describe the system architecture in Section III and trust and threat model in Section IV. Critical system operations are detailed in Section V. The evaluation setup is presented in Section VI while Section VII covers the results. Related work is summarized in Section VIII. The paper is concluded in Section IX.

II. Motivating Example

In this section, we use an application agnostic example to show how user reputation, which has long been used as a proxy to assess data trustworthiness, can inadvertently leak user privacy in the context of participatory sensing. In particular, we demonstrate the case in which reputation information is exploited by an adversary to link anonymous user contributions. Such linkages, if established over a period of time, are likely to reveal the uniqueness among travel patterns, which in turn allows the adversary to identify users. We consider a typical participatory sensing application, wherein users are requested to collect sensor data and upload them to an application server. To remain anonymous, each user identifies himself using a pseudonym, PID, and annotates the sensor data, $\bar{s}$, with the time and location coordinates, t, x, y, at which the readings are collected. For maximum anonymity, different pseudonyms are used in different time intervals. To further preserve their privacy, users cloak the embedded temporal and spatial information using techniques such as [7]. Cloaking allows users to replace the actual attribute values with the anonymized versions. Note that, we use the terms, cloaking and anonymizing, interchangeably in this paper. This ensures that users are indistinguishable from each other (by sharing the same attribute values), so that they cannot be uniquely identified. If we label the anonymized times and locations as t', x', y', then the data tuple $D_{i,t} = \langle PID_i, t'_{i,t}, x'_{i,t}, y'_{i,t}, \bar{s}_{i,t} \rangle$ denotes the data uploaded to the application server by the i^{th} user at time t.

We assume that there exists an adversary who has access to $D_{i,t}$ from all contributing users and whose goal is to uniquely identify those users. In the context of this work, we regard the application server as an instance of this type of adversary. One way in which the adversary can achieve this is by linking successive contributions from the same user. The rationale behind the linking attack is follows. While a group of users share a common t', x', y' in one time interval, they may individually use different anonymized values in subsequent intervals due to their independent motions. As such, sequences of t', x', y' submitted by different users have high probabilities of being unique, which allow the adversary to distinguish individual users. However, linking successive contributions from the same user is a non-trivial task for the attacker if pseudonyms are used. Consider an illustrative case in which the i^{th} user makes contributions in $t = 1$ and $t = 2$. Since $PID_{i,1} \neq PID_{i,2}$, the adversary possessing no additional information is unlikely to ascertain the connection between $t'_{i,1}, x'_{i,1}, y'_{i,1}$ and $t'_{i,2}, x'_{i,2}, y'_{i,2}$. However, the availability of reputation values can readily allow the adversary to establish the necessary linkages and thus de-anonymize the users. We next sketch this process.

Assuming now that, as a result of his contribution at $t = 1$, a reputation value, $r_{i,1}$, is assigned to the i^{th} user. For the purpose of linking user contributions, the adversary (which is synonymous with the application server in this context) creates a mapping between pseudonym and user reputation as $(PID_{i,1}, r_{i,1}|AS)$ and stores it in a database. In order to assure the application server of the quality of his sensor data at $t = 2$, the same user includes his reputation value in the contribution $D_{i,2}$. Let us denote this extra information as $r_{i,2}|TTP$. We explain the origin of $r_{i,2}|TTP$ as well as the different reputation suffixes in Section III. Since $r_{i,1}|AS$ and $r_{i,2}|TTP$ are computed based on the same input data (also explained in Section III), i.e., contribution of the user at time $t = 1$, hence, $r_{i,1}|AS = r_{i,2}|TTP$. This means that, by looking for a reputation value in the set of D_i at $t = 2$ which equals to $r_{i,1}|AS$, the adversary can link $PID_{i,1}$ (which is associated with $r_{i,1}|AS$ in the database) with $PID_{i,2}$ (which references the data contributor with $r_{i,2}|TTP$) and concludes that the respective t', x', y' characterize the successive movements of the exact same user.

This example highlights the danger of naively revealing the actual reputation values. It inadvertently leaks user privacy by affording the adversary the ability to track users over successive contributions, even if pseudonyms are used. In light of such danger, we are thus motivated to devise a privacy-preserving reputation system.

III. System Architecture

In this section, we describe the architecture of our system whose pictorial representation is shown in Fig. 1. It consists of three entities: (1) users (2) trusted third party (TTP) server and (3) application server (app. server). We have opted for a centralized architecture in favour of a distributed approach for the following reasons: A distributed architecture requires pre-existing trust among peers so that they can confidently exchange private information. However, in participatory sensing (cf. conventional infrastructure sensor network), such requirement is impractical since neighboring conditions, i.e., nearby users, are constantly changing and potentially short-lived. Note that, our system can readily work with any attribute-obfuscating and reputation systems. As an illustrative example, in this work, we choose those proposed in [7] and [8], respectively.

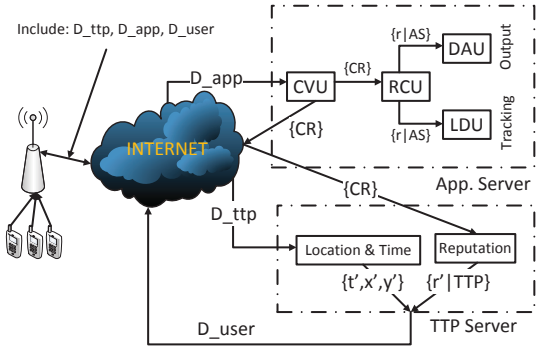

Fig. 1. System Architecture With Information Flow

The following notations are used. We assume that there are N participating users. We label the actual identity of user i as UID_i and each user replaces this with a pseudonym, PID_i, when interacting with the app. server. Note that, a different pseudonym is used for each new contribution. We denote the time and location coordinates[1] as t_i, x_i, y_i and user reputation as r_i. The sensor value collected is represented by the vector $\bar{s}_i$. Furthermore, curly braces are used to denote sets of values, e.g., $\{t, x, y\}$ refers to the set of times and location coordinates from multiple users. We use the prime notation to distinguish anonymized values, e.g., t'_i represents the anonymized time. Lastly, we use $D_{subscript}$ to label various data tuples exchanged in the system. Note that, $subscript$ is used to indicate the data recipient, e.g., D_{user} is delivered to users.

A. Data Exchanged Between User and the TTP Server

We begin by describing the data exchanged between the users and the TTP server. Users contact the TTP server for two purposes: (1) anonymizing $\{t, x, y\}$ to be included in their uploads and (2) retrieving $\{r'\}$. Each user uploads $D_{ttp,i} = \langle UID_i, PID_i, t_i, x_i, y_i \rangle$ to the TTP server. The server first anonymizes $\{t, x, y\}$ using the algorithm presented in Section V-A. This step ensures that a user becomes temporally and spatially indistinguishable from a number of other users, thus, lowering the risk of the user being uniquely identified via timestamps and coordinates [10]. The result is a set of temporal and spatial equivalence classes with each class being represented by a pair of anonymized time and location. Table I shows an example with 12 users. $(UID_1, UID_2, UID_3, UID_4)$ constitute one temporal and spatial equivalence class, which is represented by $t' = 09:08:43, x' = 0.55, y' = 0.57$. We label the j^{th} equivalence class as EC_j^{s+t} and the class representation as $t', x', y'|EC_j^{s+t}$. The server returns to users $D_{user,i} = \langle t'_i, x'_i, y'_i|EC_j^{s+t}, r'_i \rangle$, $i \in EC_j^{s+t}$. Note that, we deliberately defer the explanation pertaining to the anonymized reputation, r'_i, to Section III-C. For now, we simply assume that the server has computed and anonymized the reputation values for users. With $D_{user,i}$, users create contributions of the form $D_{app,i} = \langle PID_i, t'_i, x'_i, y'_i|EC_j^{s+t}, r'_i, \bar{s} \rangle$ and upload them to the app. server. Note that, we omit specifying EC_j^{s+t} in the rest of this paper unless unambiguity between different equivalence classes is important for the context.

UID	t	x	y	t'	x'	y'	r	r'
1	09:01:30	0.31	0.48	09:08:43	0.55	0.57	0.62	0.80
2	09:05:15	0.63	0.55	09:08:43	0.55	0.57	0.78	0.80
3	09:10:53	0.37	0.50	09:08:43	0.55	0.57	0.88	0.80
4	09:15:40	0.42	0.46	09:08:43	0.55	0.57	0.50	0.48
5	09:18:24	0.82	0.91	09:27:32	0.74	0.86	0.47	0.48
6	09:24:36	0.76	0.81	09:27:32	0.74	0.86	0.46	0.48
7	09:30:52	0.68	0.88	09:27:32	0.74	0.86	0.21	0.22
8	09:32:46	0.70	0.75	09:27:32	0.74	0.86	0.35	0.22
9	09:40:11	0.55	0.73	09:46:30	0.62	0.65	0.11	0.22
10	09:42:55	0.50	0.68	09:46:30	0.62	0.65	0.92	0.90
11	09:45:16	0.66	0.60	09:46:30	0.62	0.65	0.85	0.90
12	09:50:03	0.75	0.58	09:46:30	0.62	0.65	0.89	0.90

TABLE I
SAMPLE OF ANONYMIZED TIMES, LOCATIONS AND REPUTATION VALUES FOR 12 USERS

B. Data Exchanged Between User and the App. Server

One of the main tasks of the app. server is to compute the aggregate statistics from user contributions. In any typical participatory sensing application, the app. server is, in most cases, unaware of the sensing contexts of data contributors. For instance, it has no means of knowing whether a user has correctly oriented the embedded microphone for sampling ambient noise. A standard approach in this circumstance is to exploit the inherent redundancy in participatory sensing and resort to some form of consensus-based outlier detection technique to cross validate user data and ultimately improve the output accuracy [11] [12]. In the system shown in Fig. 1, outlier detection is performed in the cross validation unit (CVU). The input to the CVU is a set of user contributions, $\{D_{app}\}$, with the same (anonymized) temporal and spatial annotations. This ensures that the cross validation is applied on sensor readings that describe the same physical phenomenon. The CVU then applies on the input data a suitable outlier detection algorithm (an example is provided in Section V-B) and produces a set of cooperative ratings, which we denote as $\{CR\}$. Note that, CR is generated for each user whose contribution appears in the CVU input. A cooperative rating can be interpreted as the server's experience from the current interaction with a user. As demonstrated in [8], the accuracy of aggregate statistics can be further improved if user reputations are taken into account. Therefore, cooperative ratings are dispatched to the reputation calculation unit (RCU),

[1]Our system works with any localization systems such as GPS and/or Wi-Fi/cellular triangulation.

wherein user reputations are computed. User reputation is an aggregate measure for long-term trustworthiness and is the result of multiple interactions with a user. Within the RCU, the input cooperative ratings are individually passed through a reputation function, which produces a set of reputation values. We use $\{r|AS\}$ to denote the reputation values computed by the app. server. The RCU next sends $\{r|AS\}$ to the data aggregation unit (DAU), wherein the aggregate statistics are calculated. The way in which $\{r|AS\}$ are used in the DAU depends on the nature of the underlying application. We describe an example as part of our evaluations in Section VI.

C. Data Exchanged Between the TTP and App. Servers

It is clear from Section III-B that the aggregate statistics are influenced by the reputation values, which in turn are derived from the cooperative ratings. Therefore, we can increase the output accuracy by improving the precision of $\{CR\}$. This can be easily facilitated if user reputations resulting from earlier contributions were made available to the CVU [8]. For example, the CVU can make use of the extra information to filter previously untrustworthy users, so that their contributions are not included in the consensus-building process. Unfortunately, providing reputation values to the CVU is non-trivial when pseudonyms are used. Let us consider the i^{th} user who selects a pseudonym $PID_{i,t}$ at time t. Assuming that, as the result of this contribution, the RCU has computed his reputation as $r_{i,t}|AS = 0.65$ and records the $(PID_{i,t}, r_{i,t}|AS = 0.65)$ association. In his next contribution, a different pseudonym, $PID_{i,t+1} \neq PID_{i,t}$, would be chosen. Since the app. server does not know $PID_{i,t+1}$ and $PID_{i,t}$ correspond to the same user, it cannot apply $r_{i,t}|AS = 0.65$ to filter the i^{th} user at time $t+1$. Our system overcomes this problem by asking users to provide reputation values in their contributions.

In addition to sending the cooperative ratings to the RCU, the CVU returns the (PID_i, CR_i) pair to the TTP server. Note that, we choose to return $\{CR\}$ to a trusted entity rather than directly to the users since the latter may maliciously change the values to fake their reputations. Since the TTP server knows the mapping between UID and PID, it is able to act on $\{CR\}$ and compute the reputations for the corresponding users. More specifically, the TTP server produces a reputation value for each input user by processing $\{CR\}$ individually using the same reputation function as that adopted by the RCU in the app. server. We distinguish the reputations maintained at the TTP server as $\{r|TTP\}$ (cf. $\{r|AS\}$ maintained at the app. server). Before returning $\{r|TTP\}$ to the users, the TTP server anonymizes these reputation values. As we show in Section IV, this step is crucial in preventing the inadvertent leakage of user privacy. The result is a set of reputation equivalence classes with each class being represented by a common reputation value. We similarly denote the j^{th} equivalence class as EC_j^r and the class reputation as $r'|TTP, EC_j^r$. An example is presented in Table I which includes 4 reputation equivalence classes. Once user reputations are computed and anonymized, the TTP server returns to users $D_{user,i} = \langle t'_i, x'_i, y'_i|EC_j^{s+t}, r'_i|TTP, EC_j^r\rangle$. Since anonymizations are performed separately on $\{t, x, y\}$ and $\{r\}$, the compositions of EC_j^{s+t} and EC_j^r would be different. With their temporal, spatial and reputation attributes properly protected, users upload the data tuple $D_{app,i} = \langle PID_i, t'_i, x'_i, y'_i|EC_j^{s+t}, r'_i|TTP, EC_j^r, \bar{s}\rangle$ to the app. server.

IV. Trust and Threat Models

We assume that the users have the appropriate program installed on their devices to collect data from on-board sensors. Furthermore, users are assumed not to alter any readings generated by their devices. This can be enforced by using trusted computing technique, e.g., [13]. We also assume that the TTP server computes a message digest for $\{t', x', y', r'|TTP\}$ and signs them with its private key, so that any changes can be detected by the app. server. The TTP server is the central entity for safeguarding user privacy and is assumed not to disclose the following information to an adversary - (1) the relationship between UID and PID (2) the mapping between actual and anonymized values for any attributes and (3) the anonymization parameters (to be introduced in Section V).

While user reputations can be used to evaluate data trustworthiness, they would also inadvertently leak user privacy to the app. server. We herein declare the app. server as the target threat. Such threat model is not difficult to materialize in our context. For instance, the open nature of participatory sensing allows an attacker to easily deploy a sensing application which may well appear as completely legitimate. This allows him to monitor users' $\{t', x', y'\}$ attributes, which eventually leads to the de-anonymization of users. In what follows, we explain how the app. server utilizes reputation values to breach user privacy. We have described in Section II the case wherein consecutive uploads (albeit labeled with different pseudonyms) from the same user can be linked if actual reputation values were revealed. We now focus on the other case in which anonymized reputation values, $\{r'|TTP\}$, are published.

The link discovery attempt is made in the link discovery unit (LDU) depicted in Fig. 1. Let us consider the i^{th} user at time t. Assume that as the result of his contribution, the CVU assigns to him a cooperative rating $CR_{i,t}$ and based on which, the RCU computes a reputation value $r_{i,t}|AS$. The app. server records $D_{ldu,i} = \langle PID_{i,t}, x'_{i,t}, y'_{i,t}, r_{i,t}|AS\rangle$ in its database. At the same time, the app. server also returns $CR_{i,t}$ to the TTP server as described previously. Based on this information, the TTP server updates the user's reputation and anonymizes it as $r'_{i,t+1}|TTP$. Note that, the subscript $t+1$ is used to emphasize the fact that this value would be retrieved by the user for his next contribution. At time $t+1$, the app. server receives $\langle PID_{i,t+1}, x'_{i,t+1}, y'_{i,t+1}, r'_{i,t+1}|TTP\rangle$. Without anonymization, $r_{i,t+1}|TTP = r_{i,t}|AS$ and the equality between $PID_{i,t}$ and $PID_{i,t+1}$ would be easily established. However, since $r'_{i,t+1}|TTP \neq r_{i,t}|AS$ due to anonymization, hence, additional processing is required to deduce the link between the two pseudonyms. In this work, we assume that the app. server applies a simple filtering technique as follows. For each $D_{ldu,i}$ in its database, the app. server calculates the Euclidean distance, $\mathcal{L}_{i,q}$, between $(x'_{i,t}, y'_{i,t})$ and $(x'_{q,t+1}, y'_{q,t+1})$ where $q = 1 \ldots N$ and selects those $\mathcal{L}_{i,q} \leq \epsilon$ as the list of

users whose pseudonyms potentially reference the same user as $PID_{i,t}$. The distance filtering is implemented in light of the fact that users' movements are constrained by their modes of transportation. For example, a taxi operating in a city during rush hour is unlikely to travel $60km/hr$. Note that, the filtering threshold ϵ is application-specific and can be estimated by analyzing historical user mobility patterns. The size of the resulting list will be used in our evaluations as a metric for linkability.

V. System Operations

In this section, we provide further details on the key system operations previously described in Section III. In particular, we elaborate on the anonymization performed by the TTP server, provide an example of outlier detection algorithms and introduce the reputation function used by the TTP and app. servers.

A. Attribute Anonymization

We first present the algorithm used by the TTP server to anonymize temporal, spatial and reputation information for users. The algorithm is based on the results from [7]. In particular, the variable-length, maximum distance to average vector (V-MDAV) algorithm [14] is used. The advantage of V-MDAV is that it works particularly well with numerical attributes. V-MDAV requires two parameters. The first parameter, k, specifies the degree of anonymity, i.e., how many users share a common anonymized value, while the second parameter, Φ, measures the similarity among users. In contrast to [7], wherein temporal and spatial similarities are separately calculated, we herein use a composite metric $\Phi_{cps} = \Phi_t + \Phi_{x,y}$, where $\Phi_{x,y}$ is simply the Euclidean distance between two pairs of GPS coordinates and Φ_t is the absolute difference between two GPS times. The composite metric is used here so that the app. server can group co-located users who contribute at similar times for processing. The similarity among user reputations is measured by Φ_r, which equates to the absolute difference between two reputation values.

The TTP server takes $\{t, x, y\}$ and (k_{s+t}, Φ_{cps}) as inputs to V-MDAV, which produces a set of equivalence classes, $\{EC^{s+t}\}$, with each j^{th} class accommodating at least k_{s+t} users who share a common anonymized time and location. The algorithm also guarantees that members of each EC_j^{s+t} are closest in terms of time and location, i.e., having smallest Φ_{cps}. Table I shows a sample of anonymized times and locations for 12 users with $k_{s+t} = 4$. Similarly, the set of reputation values, $\{r|TTP\}$, and (k_r, Φ_r) are supplied to V-MDAV, which outputs a set of equivalence classes, $\{EC^r\}$, with each j^{th} class containing k_r users who share a common reputation value, $r'|TTP, EC_j^r$. A sample is also shown in Table I with $k_r = 3$. Note that, different subscripts are affixed to k to highlight that the anonymization of time and location is performed independently of reputation. Such decoupling is based on the fact that it is plausible for users in different locations to possess similar reputations. It is also worth mentioning that the above procedures apply to operations in one time interval. Since users move independently of each other and their contributions are often of disparate quality (which causes their reputations to vary differently), the compositions of $\{EC^{s+t}\}$ and $\{EC^r\}$ would be different in each time interval.

B. Outlier Detection

We next provide an example of outlier detection algorithms used by the CVU to produce cooperative ratings. The example algorithm is intended to work with applications that compute average sensor values. It is based on the iterative algorithm originally proposed in [15] to compute the robust average values in a mote-based sensor network. As the iterations converge, the weights assigned to individual sensor readings are taken as the cooperative ratings for the input users. It was shown in [8] that this algorithm benefits greatly if prior reputations were used to preemptively eliminate disreputable contributors, so that their corrupted sensor data do not propogate through the processing pipeline. Therefore, we herein adopt the same reputation feedback approach and examine if revealing anonymized reputations, $\{r'|TTP\}$ (recall that the app. server obtains previous reputation values from users), would affect the computation of cooperative ratings and ultimately degrade the accuracy of outputs computed by the DAU.

C. Reputation Function

We now briefly describe the function used to compute user reputations. A detailed discussion on the choice of reputation function is provided in [8]. In this work, user reputations are modeled by Gompertz function [17] whose mathematical construct is shown below,

$$r = e^{be^{c \times (\sum_{t'=1}^{t} \lambda^{t-t'} \times CR_{t'})}} \quad (1)$$

The parameters b and c control the growth rate of the function. The function input is a time-weighted sum of cooperative ratings. The summation indicates that historical experiences are considered while the weighting is applied to highlight the most recent contribution and discount distant ones. In addition, different weighting factors are used so that user reputations are slowly accumulated (as results of cooperative contributions) but quickly destroyed (as results of non-cooperative contributions). The asymmetric rates closely model the trust among humans (i.e., the primary sensing entity in participatory sensing) in social interactions, e.g., we often slowly build our trust towards others after several instances of good behavior but rapidly tear down the trust if experiencing only a handful of dishonest behavior. The range of function output, i.e., user reputation, is between 0 and 1.

VI. Evaluation Setup

We conduct several simulations to evaluate the performance of our system. Our evaluations have two objectives - (1) quantifying how likely it is for an adversary to link contributions from the same users over time (the linkability objective) and (2) measuring the amount of degradations in output accuracy as the results of anonymizing reputation values (the accuracy objective).

A. Example Application

We evaluate our system by incorporating it within a real-world participatory sensing application. We consider a noise monitoring application similar to [4], which relies on volunteers to collect ambient noise level using their mobile phones. In such application, a noise monitoring client is installed on the user's phone. The client intelligently detects if the phone is exposed to open spaces (using other on-board sensors) and samples the ambient noise level at a specified frequency. Each noise sample is annotated with the user's anonymized time, t', coordinates, (x', y'), and reputation values, $r'|TTP$, and then submitted to the app. server via 3G or Wi-Fi networks. Upon receiving noise samples, the app. server groups those with the same t', x', y' and dispatches the results to the CVU, wherein cooperative ratings, $\{CR\}$, are produced by executing the robust average algorithm introduced in Section V-B. $\{CR\}$ are sent to the TTP server as per Section III-C as well as passed down to the RCU, wherein reputation values, $\{r|AS\}$, are generated using the Gompertz function. The resulting reputation values are separately forwarded to the DAU and LDU for computing application outputs and linking user contributions, respectively. Within the DAU, reputation values act as weighting coefficients to compute the noise levels at all locations specified by $\{(x', y')\}$. The LDU proceeds as described in Section IV to discover the relationships among successive user uploads.

B. Dataset

We evaluate our system in real-world deployment scenarios. For this, we use the mobility traces from [19] to model the travel patterns of typical urban users. The dataset contains the GPS timestamps and coordinates of approximately 500 taxis collected in May, 2008 in the San Francisco Bay Area. This dataset is chosen since the underlying entity (a network of taxis) integrates nicely with the above application. For example, a city council may mandate all taxis to collect noise readings for the purpose of monitoring noise pollution in a city[2], due to their wide coverage of the city landscape.

For simplicity, we only use a part of the traces in our simulations. Specifically, traces between 16:00 and 19:00 on 19/05/2008, which contain 150 taxis, are selected. We assume that all users actively contribute sensor readings over the entire simulation period. Further, we assume that each user reports once every 5 minutes with each contribution containing 60 seconds of noise data sampled at 1-second granularity. Since each user contribution triggers a new reputation calculation, we thus have 37 reputation updates per simulation.

C. Simulation Model

Let us now describe the simulation model. Each simulation involves running the example application in the sequence described in Section III. At the start of simulation, each (of the 150) user is assigned GPS timestamp and coordinates, t, x, y, taken from the taxi traces[3]. The TTP server executes V-MDAV on all $\{t, x, y\}$ received with parameters k_{s+t} and produces the corresponding anonymized versions $\{t', x', y'\}$. The TTP server also computes and anonymizes (by applying V-MDAV with parameter k_r) reputation values for users. For the very first interaction, the TTP server simply assigns users with some initial reputations while subsequent reputation values are calculated as per Section III-C. With $\{t', x', y', r'\}$ returned, users proceed to generating noise values, $\{\bar{s}\}$. Since we do not have enough resources to survey the noise distribution in an urban environment, we synthesize such values by assuming the noise processes follow a normal distribution with mean μ and standard deviation σ. We also introduce unreliable users into the simulation to reflect a more realistic usage scenario. The number of unreliable users in a temporal and spatial equivalence class, EC_j^{s+t}, is approximated by a normal distribution with mean μ_u and standard deviation σ_u. We further assume that these unreliable users are independent, i.e., they add different offsets to the raw noise readings. While the above represent simplified approximations and assumptions, we justify the decisions by noting that, they do not invalidate the Gompertz function and the resulting transitions of reputation values. With $\{t', x', y', r', \bar{s}\}$ at hand, users prepare the data tuple $\{D_{app}\}$ and send them off to the app. server, which processes the received data following the procedures discussed earlier in Section VI-A.

D. Evaluation Metrics

We now introduce the metrics used in the evaluations. Note that, the following descriptions apply to one reputation update interval. Recall in Section IV, we discussed a distance filtering technique for the app. server to link user contributions when they present anonymized reputation values. If we denote the resulting list of users qualifying for $PID_{i,t} = PID_{q,t+1}$ as Ψ_i, the linkability metric is define as $\frac{1}{|\Psi_i|}$. To measure the degradations in output accuracy, we compare the application outputs against the ground truth. The ground truth for the noise application is ideally recorded at the center of the sampling region with a sound level meter. However, as our evaluations are based on synthetic data, the ground truth is also synthetically generated by using the same distribution specified in Section VI-C. We compare the average noise levels calculated from user contributions against the ground truth using the root-mean-squared-error (RMSE) criterion. The RMSE between two vectors of values is defined as Eq. 2

$$RMSE = \sqrt{\frac{\sum_{m=1}^{T}(\bar{v}_{1,m} - \bar{v}_{2,m})^2}{T}} \tag{2}$$

In our evaluations, $T = 60$ since each user contribution contains 60-seconds worth of noise data. The RMSE is calculated separately for each temporal and spatial equivalence class.

VII. Evaluation Results

In this section, we discuss the results from our simulations. Section VII-A presents the findings in regards to the linka-

[2]We assume that digital signal processing techniques are used to separate urban background noise from (undesirable) conversations within the taxis.

[3]In cases where multiple GPS updates exist in an update interval, only the first one is selected.

bility objective (see Section VI) while those pertaining to the accuracy are given in Section VII-B.

A. Comparisons of Linkability

Each simulation is conducted according to the model described in Section VI-C. Table II summarizes the corresponding simulation parameters. Note that, the number of unreliable

par	*val*	*par*	*val*
k_{s+t}	10	b	-2.10
k_r	10	c	-0.45
μ	60	λ_1	0.70
σ	5	λ_2	0.8
μ_u	$0.5 \times \lvert EC_j^{s+t} \rvert - 1$	ϵ	0.5
σ_u	1		

TABLE II
SIMULATION PARAMETERS

users in each temporal and spatial EC is a function of the class size. The parameter, ϵ, specifies the reputation threshold for removing disreputable users as per Section V-B. Two aging factors, λ_1 and λ_2, are respectively applied to cooperative and non-cooperative users (see Section V-C). Recall that, all users are assumed to actively contribute in every reputation update interval, thus, the app. server knows that for each $D_{app,i}$ at time $t+1$, there must have been a contribution from the same user (albeit under a different pseudonym) at time t. To establish the connection, the app. server applies the distance filtering technique described in Section IV. The resulting linkability is calculated for each user in every but the first update interval. We repeat the simulation for 100 times and plot the per-user average probabilities in Fig. 2.

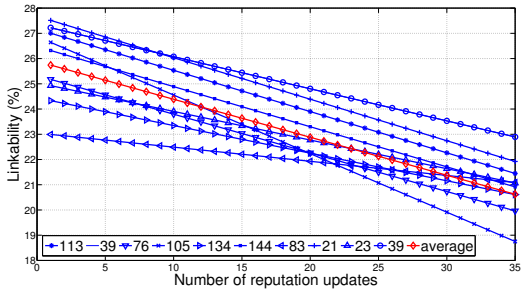

Fig. 2. Average Linkability for Selected Users

The data shown in Fig. 2 have been linearly fitted to better reveal the trend over time. Due to the sheer number of users in the dataset, we only show the linkability for a subset of them as well as the values averaged over all users. Recall in Section II, we show that an adversary can virtually track each and every person, i.e., 100% linkability, via the connections inherent in user reputations. With our anonymization scheme in place, the average probability of a successful linkage is reduced to around 25% at the beginning of data contribution. As time progresses, the average probability continues to decrease and we observe a 21% average linkability at the end of the observation period. This is equivalent to a phenomenal 79% average improvement. To explain the declining trend in Fig. 2, we need to remember that the distance filtering technique works on location coordinates in successive time intervals. In other words, if the adversary made a wrong linkage between contributions in time t and $t+1$, the error in the spatial information would propagate and compound to that at $t+2$, which makes it increasingly difficult to track users.

B. Comparisons of Output Accuracy

We next present the results from the accuracy simulations. Each simulation again follows the model described in Section VI-C and assumes the parameter values in Table II. For each reputation update interval, we average the RMSEs over all equivalence classes. We then repeat the simulation 100 times and plot the results in Fig. 3. We can see

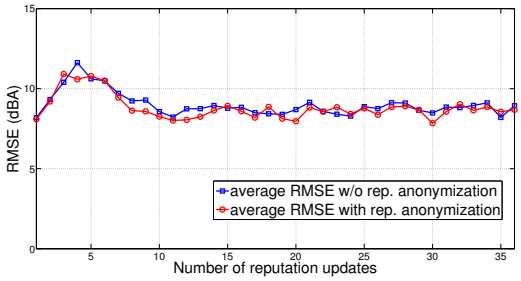

Fig. 3. Average Root-Mean-Squared-Errors in Estimating Noise Levels

that anonymizing user reputations does not cause discernible degradations in output accuracy. To understand the reason behind this, we must re-examine the relationship between $\{r|TTP\}$ and $\{r'|TTP\}$. As stated in Section V-A, one of the most important features of V-MDAV is that the differences between member reputation values, $\{r|TTP\}$, and their class representation, $\{r'|TTP, EC_j^r\}$, are made as small as possible. In other words, if a user with $r_i|TTP \geq \epsilon$ was not filtered by the CVU then, he would also be trusted by revealing $r_i'|TTP$. Exceptions occur when $r_i|TTP$ is close on either side of the filtering threshold, e.g., $r_i|TTP = 0.46$ could be anonymized to $r_i'|TTP = 0.52$ causing the CVU to pass the corresponding user contribution down the processing pipeline if $\epsilon = 0.5$. This means that in most cases, reputation anonymization does not change the set of user contributions that participate in the calculation of weighted averages at the DAU. The initial spike in RMSEs can be attributed to the fact that reputation, being a long term measurement, requires time to learn before it reaches a steady state [8].

Combing the above results with those from Section VII-A, we can see that our anonymization scheme ensures user privacy (by reducing the linkability) while maintaining similar levels of application output accuracy.

VIII. Related Work

The problem of preserving user privacy while ensuring data trustworthiness has rarely been explored in the context of participatory sensing. Krontiris acknowledged the importance of a privacy-preserving reputation system in [20] but left the actual implementation as a future work. Privacy-preserving reputation systems, however, have been researched extensively in peer-to-peer networks. For example, [21] proposed the use of trusted computing technology for providing distributed

reputation system with privacy. Voss described in [22] two cryptographic-based reputation schemes to dissolve the links between pseudonyms for mobile information dissemination networks. A common feature among these works is that, privacy is provided through the unlinkability among pseudonyms. However, they do not consider the linkability exposed by reputation values as shown in Section II. While neglecting such aspect may be harmless in peer-to-peer networks (since the probability of meeting the same peer in a large scale deployment is remote), it is detrimental in our context as the app. server is in constant contact with the users and is able to observe the transitions in user reputations. In this regard, the solutions presented in [9], [23], [24], [25] are most similar to ours. [9] proposed an anonymous reputation system which securely transfers reputation values from one pseudonym to the next. The authors also suggested converting user reputations to coarser granularity but stopped short at explaining how it is done. The system proposed in [23] leverages cryptographic primitives to mask the links between pseudonyms and more importantly, each pseudonym does not reveal the actual reputation but declare his membership to a particular reputation group. While it is similar to our k-anonymous based solution, computationally expensive cryptographic operations are required and can be a limiting factor for mobile devices, which still suffer from mediocre battery lifetime. Wei described a similar k-anonymous solution in [24]. However, it requires a communication path to be established among peers to share reputation information, which is considered too intrusive in our context, wherein prior trusts among users are often non-existent. In [25], the authors addressed a similar problem in the context of participatory sensing. Their solution is based on cryptographic primitives and cloaking of reputation values. A common theme of their cloaking algorithms is the partitioning of user reputations into a pre-defined set of values. Such approach lacks the flexibility of V-MDAV. More importantly, it does not guarantee the closeness among users sharing the same values, which helps to minimize accuracy degradations.

IX. Conclusion

In this paper, we identified the challenges that arise when privacy and data trustworthiness requirements need to be simultaneously met in the context of participatory sensing. We showed that users are vulnerable to linking attack if they naively reveal their reputations to the app. server. We then proposed a reputation anonymization scheme to minimize the risk of such attack. Our solution was evaluated by simulating a real-world participatory sensing application with real-world user mobility patterns. The results showed that it can reduce the linkability by as much as a factor of 6 while incurring negligible degradations in application output accuracy.

References

[1] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory Sensing", in *Proc. of WSW, in conjunction with ACM SenSys'06.*, Boulder, CO, USA, November 2006.

[2] E. Paulos, R. Honicky, and E. Goodman, "Sensing atmosphere", in *Proc. of the Workshop on Sensing on Everyday Mobile Phones in Support of Participatory Research, in conjunction with ACM SenSys'07*, 2007.

[3] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "Noisetube: Measuring and mapping noise pollution with mobile phone", in *ITEE 2009 - Information technologies in Environmental Engineering*, Springer Berlin Heidelberg, May 2009.

[4] R. Rana, C.T. Chou, S. Kanhere, N. Bulusu and W. Hu, "Ear-Phone: An End-to-End Participatory Urban Noise Mapping System", in *Proc. of IEEE/ACM IPSN '10*, 2010.

[5] S. Eisenman, E. Miluzzo, N. Lane, R. Peterson, G. Ahn and A. Campbell, "The Bikenet Mobile Sensing System for Cyclist Experience Mapping", in *Proc. of ACM SenSys'07*, 2007.

[6] Y. Dong, S. S. Kanhere, C. T. Chou and N. Bulusu,"Automatic Collection of Fuel Prices from a Network of Mobile Cameras", inn *Proc. of IEEE DCOSS'08*, 2008.

[7] K. L. Huang, S. S. Kanhere and W. Hu, "Preserving Privacy in Participatory Sensing Systems", in *Computer Communications*, vol. 33, no. 11, pp. 1266-1280, July 2010.

[8] K. L. Huang, S. S. Kanhere and W. Hu, "Are You Contributing Trustworthy Data? The Case for a Reputation System in Participatory Sensing", in *Proc. of ACM MSWiM'10)*, 2010.

[9] H. Miranda and L. Rodrigues, "A Framework to Provide Anonymity in Reputation Systems", in *Proc. of MobiQuitous'07*, 2007.

[10] L. Sweeney, "k-anonymity: A Model for Protecting Privacy," in *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557-570, 2002.

[11] M. M. Breunig, H.-P. Kriegel, R.T. Ng, and J. Sander, "LOF: Identifying Density-based Local Outliers", in *Proc. of the ACM SIGMOD Conference*, 2010.

[12] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "LOCI: Fast Outlier Detection Using the Local Correlation Integral", in *Proc. of IEEE ICDE'03*, 2003.

[13] A. Dua, N. Bulusu, W. Feng, and W. Hu, "Towards Trustworthy Participatory Sensing", in *Proc. of HotSec'09)*, 2009.

[14] A. Solanas, and A. Martinez-Balleste, "V-MDAV: a multivariate microaggregation with variable group size", in *17th COMPSTAT Symposium of the IASC*, Rome, 2006.

[15] C. T. Chou, A. Ignjatovic, and W. Hu, "Efficient Computation of Robust Average in Wireless Sensor Networks using Compressive Sensing", Technical Report: UNSW-CSE-TR-0915. ftp://ftp.cse.unsw.edu.au/pub/doc/papers/UNSW/0915.pdf

[16] S. Ganeriwal and M. Srivastava, "Reputation-based Framework for High Integrity Sensor Networks", in *ACM TOSN*, Vol. 4, No. 3, May 2008

[17] J. F. Kenney and E. S. Keeping, *Mathematics of Statistics Part 1*, 3rd ed. Princeton, NJ: Van Nostrand, 1962

[18] SPL Graph. An Audio Level Chart Recorder for the iPhone and iPod Touch, http://www.studiosixdigital.com/leq_graph.html

[19] M. Piorkowski, N. Sarafijanovoc-Djukic, and M. Grossglauser, "A Parsimonious Model of Mobile Partitioned Networks with Clustering", in *The proc. of COMSNETS'09*, 2009.

[20] I. Krontiris and N. Maisonneuve, "Participatory Sensing: The Tension Between Social Translucence and Privacy", in *Trustworthy Internet*, 2011, pp. 159-170.

[21] M. Kinateder and S. Pearson, "A Privacy-Enhanced Peer-to-Peer Reputation System", in *E-Commerce and Web Technologies*, 2003, vol. 2738, pp. 206-215.

[22] M. Voss, A. Heinemann, and M. Muhlhauser,"A Privacy Preserving Reputation System for Mobile Information Dissemination Networks", in *Proc. of SecureComm'05*, 2005.

[23] E. Androulaki, S. G. Choi, S. M. Bellovin, and T. Malkin, "Reputation Systems for Anonymous Networks", in *Privacy Enhancing Technologies*, 2008.

[24] Y. Wei and Y. He, "A Pseudonym Changing-based Anonymity Protocol for P2P Reputation Systems", in *Proc. of ETCS'09*, 2009.

[25] D. Christin, C. Robkopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, "IncogniSense: An Anonymity-preserving Reputation Framework for Participatory Sensing Applications", in *Proc. of IEEE PerCom'12*, 2012.