ARTSense: Anonymous Reputation and Trust in Participatory Sensing

Xinlei (Oscar) Wang*, Wei Cheng*, Prasant Mohapatra*, Tarek Abdelzaher†

*Department of Computer Science
University of California, Davis, CA 95616
{xlwang, weicheng, pmohapatra}@ucdavis.edu
†Department of Computer Science
University of Illinois at Urbana Champaign, Urbana, IL 61801
zaher@cs.uiuc.edu

***Abstract*—With the proliferation of sensor-embedded mobile computing devices, participatory sensing is becoming popular to collect information from and outsource tasks to participating users. These applications deal with a lot of personal information, e.g., users' identities and locations at a specific time. Therefore, we need to pay a deeper attention to privacy and anonymity. However, from a data consumer's point of view, we want to know the source of the sensing data, i.e., the identity of the sender, in order to evaluate how much the data can be trusted. "Anonymity" and "trust" are two conflicting objectives in participatory sensing networks, and there are no existing research efforts which investigated the possibility of achieving both of them at the same time. In this paper, we propose *ARTSense*, a framework to solve the problem of "trust without identity" in participatory sensing networks. Our solution consists of a privacy-preserving provenance model, a data trust assessment scheme and an anonymous reputation management protocol. We have shown that ARTSense achieves the anonymity and security requirements. Validations are done to show that we can capture the trust of information and reputation of participants accurately.**

I. Introduction

In recent years, we have seen the massive prevalence of mobile computing devices such as smartphones and tablet computers. These devices usually come with multiple embedded sensors, such as camera, microphone, GPS, accelerometer, digital compass and gyroscope. Because of these advancements, the participatory sensing model is becoming popular. Participants use their personal mobile devices to gather data about nearby environment and make them available for large-scale applications. Two examples of participatory sensing applications are Gigwalk [1] developed by a startup company and mCrowd [2] developed by University of Massachusetts Amherst. They provide a marketplace for sensing tasks that can be performed from smartphones. A requester of data can create tasks that uses the general public to capture geo-tagged images, videos, audio snippets, or fill out surveys. Participants who have installed the client apps on their smartphones can submit their data and get rewarded. For example, Microsoft Bing has been collecting photos using Gigwalk for panoramic 3-D photosynthesis of businesses and restaurants in Bing Map.

Sharing sensed data tagged with spatio-temporal information could reveal a lot of personal information, such as user's identity, personal activities, political views, health status, etc. [3], which poses threats to the participating users. Therefore, participatory sensing requires a deeper attention to privacy and anonymity, and a mechanism to preserve user's location privacy and anonymity is mandatory. Another dimension of data security in participatory sensing is the reliability of the sensed data. In participatory sensing applications, data originates from sensors controlled by other people, and any participant with an appropriately configured device can easily submit falsified data, hence data trustworthiness becomes more crucial than the traditional wireless sensor networks. There is an inherent conflict between trust and privacy. If a participatory sensing system provides full anonymity to the participants, it is difficult to guarantee the trustworthiness of submitted data. Finding a solution that achieves both trust and anonymity is a major challenge in such systems [4].

There have been plenty of research efforts that have investigated privacy techniques for anonymous data collection in location based services (LBS) and particularly in participatory sensing systems. However, how to assess the trustworthiness of the anonymously collected data has not been considered. Other pieces of work which studied trust models did not consider the privacy requirements. In this paper, we are trying to solve the problem of "trust without identity" in participatory sensing networks. To the best of our knowledge, our work is the first attempt for a trust and reputation framework while maintaining the desired anonymity in the context of participatory sensing. To summarize, the **contributions** of our work include:

1) A novel provenance model for participatory sensing applications is developed which serves as the basis of sensing data trust assessment while maintaining the appropriate level of user anonymity.
2) A trust assessment algorithm is proposed to compute the trust of sensing reports based on anonymous user reputation levels and privacy-preserving contextual factors such as location, time, sensor mode and traveling mode.
3) An anonymous reputation management mechanism is presented to maintain the anonymity properties while also enforce positive or negative user reputation updates.
4) Analytical and empirical validations are done to show our ARTSense scheme achieves the anonymity and security objectives, and captures both user reputation and data trust accurately.

The rest of the paper is organized as follows. We highlight the related work of data security in participatory sensing in Section II. In Section III, we give an overview of the system model including a formal definition of trust and reputation. The threat model will also be detailed in this section. We then

present our proposed ARTSense scheme in Section IV. The security analysis of our scheme is given in Section V and performance evaluations based on simulation are presented in Section VI. We give a discussion and talk about our future work in Section VII. Finally, Section VIII concludes the paper.

II. Related Work

Privacy preserving techniques have been extensively studied in the context of LBS. A group of well-known techniques in preserving user privacy is the spatial and temporal cloaking technique [5], [6], where the participant's location at a specific time is blurred in a cloaked area or cloaked time interval, while satisfying the privacy requirements. Most of these techniques are based on k-anonymity [7], where the location of a user is cloaked among $k-1$ other users.

In addition to the studies about privacy in the context of LBS, a few pieces of recent work [8]–[10] have specifically studied the privacy in participatory sensing. In [8], the concept of participatory privacy regulation is introduced. In [9], [10], different approaches are proposed, which focus on with how participants upload the collected data to the server without revealing their identity. Our work is different from these works in that we are trying to solve the problem of "trust without identity" in participatory sensing networks instead of the process of anonymous data collection.

There have been numerous trust systems proposed toward the data reliability in ad hoc networks, traditional wireless sensor networks and participatory sensing networks as well, for example, [11], [12]. However, none of these approaches considered the high requirement for privacy and anonymity in the context of participatory sensing.

To address the problem of "trust without identity", anonymous reputation systems in P2P networks have been proposed [13], [14]. These systems are based on pseudonyms and eCash: an electronic cash system aims at offering anonymity properties by making spending and withdrawal unlinkable. One of the drawbacks of these protocols is that negative reputation updates are not supported. More importantly, these approaches mainly focused on dealing with anonymous mutual ratings between two interaction users in P2P networks, which cannot be applied directly to the participatory sensing applications. Our paper proposes a novel anonymous reputation management protocol specifically for participatory sensing networks and both positive and negative reputation updates can be enforced.

III. Fundamental Framework

A. System Architecture

Different participatory sensing applications may have different system models. To make it more specific, we consider a typical participatory sensing architecture, which is used by Gigwalk and mCrowd. This architecture is illustrated in Fig. 1. First of all, applications are distributed to the participants' mobile devices through App Store or other application marketplaces (Step 1). Data consumers (such as Microsoft Bing in our example) can create sensing tasks and data requirements (Step 2), and then distribute them to the mobile phones in the vicinity of the site of interest (Step 3). The sensing data collected by the phones of participants are reported (through WiFi or cellular networks) to a central application server (hereafter referred to as the "server") (Step 4). On the server,

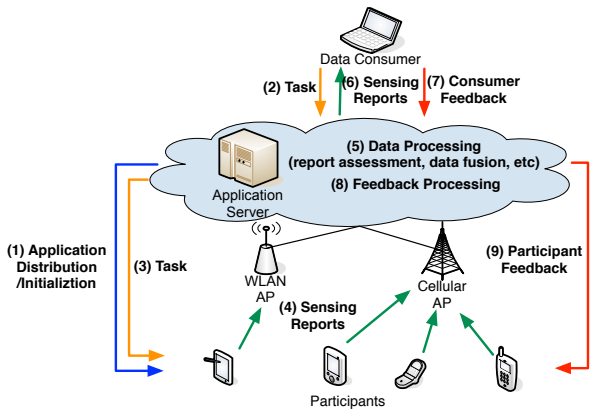


Fig. 1: Architecture of a participatory sensing system

the data are analyzed, processed (Step 5) and made available to the data consumers (Step 6). The data consumer may give feedback (e.g., credit, service fees, etc.) to the server (Step 7). Finally, the server will process the feedback (Step 8) and also give feedback (either rewards or penalties) to the participants (Step 9). Data consumers' trust and privacy is not under our consideration, so we think of them as a part of the server instead of a separated party. In particular, we will focus on what needs to be sent in Step 4, how trust assessment can be done in Step 5, the reputation feedback polices and mechanism in Step 9, and most importantly, how participants' privacy is protected in the whole process.

At the communication level of the network system, we assume a suitable anonymous network such as Onion Routing and Mix networks is applied to offer the desirable privacy protection. At the application level, we assume spatial and temporal cloaking techniques are applied to allow participants to adjust time/location resolution for individual reports. The details of how these techniques can be used have been discussed extensively and they are out of the scope of this paper.

B. Definitions of Trust and Reputation

We use the term "trust" to represent the level of confidence about the reliability and correctness of the reported sensing data. Another crucial part of the system is reputation management, including reputation demonstration and reputation update. "Trust" and "reputation" are often used interchangeably in a network trust or reputation model. We follow the definitions in [12] and use them as separated concepts. Trust is a value associated with the reported sensing data and reputation is a value associated with the participants. In addition, for privacy protection purpose, we introduce a new term "reputation level" in contrast to "reputation".

DEFINITION 1: **Trust of Sensing Reports:** The trust of a sensing report r, denoted as $T(r)$, is the probability of r being correct, as perceived by the server.

DEFINITION 2: **Reputation of Participants:** The reputation of a participant P_i, denoted as $R(P_i)$, is the synthesized probability that the past sensing reports sent by P_i are correct, as perceived by the server. The server maintains a reputation database which has the ID of each participant and the corresponding reputation. When a new participant registers with the server, the server creates a unique ID and initializes an initial reputation R_0 for the new participant in the reputation database. R_0 can be set as a value in [0, 0.5], so that newcomer

attackers can maximally get a neutral reputation.

DEFINITION 3: **Reputation Level of Participants:** The reputation level of a participant P_i, denoted as $\hat{R}(P_i)$, is a discrete approximation of reputation generated by the server based on $R(P_i)$ and granted to the participant P_i. It is used by P_i to demonstrate his/her reputation to the server without revealing his/her accurate reputation. An example of mapping $R(P_i)$ of 8.15 to $\hat{R}(P_i)$ would be rounding off the decimal and getting a result of 8. A backward mapping from $\hat{R}(P_i)$ to $R(P_i)$ should be impossible.

C. Threat Model

For the server side, we consider the server not trustworthy for protecting participants' privacy. Any information learned by the server might be leaked to a malicious server administration personnel behind the server. However, we assume the server can be trusted in terms of its functionality, e.g., user registration, key management, issuing credentials, trust assessment and reputation management. As we described in Section III-A, we assume spatial and temporal cloaking techniques are applied so that each individual sensing report is at least k-anonymous to the server. Nevertheless, if the reports submitted by a participant are linkable, e.g., the same pseudonym is used, the attacker can profile and analyze the location traces, which could reveal the identity of the sender or at least significantly reduce the possible anonymity set [3].

For the participants side, we allow anyone with an appropriate device that gets the application installed to register as a participant. An existing participant is free to abandon his/her account and register himself/herself as a new user (newcomer attack). A registered participant has the right to refuse to provide any real-identity information or accurate location and time in the sensing reports. A misbehaving participant may produce false sensing data or send false data randomly with certain probability or for certain talks (on-off attacks). An adversary may also exploit to gain unfair reputation or lie about his/her reputation level. Furthermore, we allow multiple adversaries to collusively send the same false data to deceive the server, but we assume majority of the reports are good.

We assume user authentication is done properly when the communication between a participant and the server does not need to be anonymous. Attacks via the communication channels and DoS attacks (e.g., eavesdropping, traffic jamming, etc.) are out of the scope of this paper.

IV. The ARTSense Scheme

The name of our scheme "ARTSense" indicates that we aim to achieve three objectives - "**A**nonymity", "**R**eputation" and "**T**rust" - in participatory sensing. The entire framework consists of three components: *provenance model*, *sensing report trust assessment* and *anonymous reputation management*.

A. Provenance Model

A sensing report consists of two parts, namely the payload and the provenance. The payload could be any format of sensing data, e.g., text, voice, picture, video, etc. The provenance is meta-data that describes the origin of the report, which is assumed to be automatically generated by a trusted middleware. We divide the provenance into two parts: *user provenance* and *contextual provenance*. Figure 2 illustrates the structure of a sensing report and our provenance model.

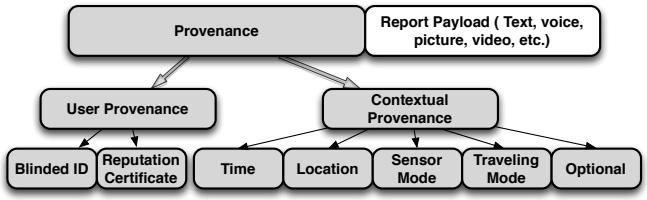


Fig. 2: Structure of a sensing report

1. User Provenance: Considering anonymity, a participant's ID should not be in the user provenance so that no one including the server can associate the participant's identity with the other information in the report. Instead, participants need to put their *Blinded ID* (BID) in the user provenance. A participant's BID acts like a pseudonym and could change randomly with every sensing report. In addition, a *Reputation Certificate* (RC) needs to be included. It is a certificate granted by the server which contains the sender's reputation level and is signed by the server. In fact, each RC is a RC pair, where one contains the user ID and the other does not. Here in the user provenance, the RC without the user ID is the one we are including. The participants demonstrate their reputation levels to the server via this anonymous RC. The reputation level is used as one of the factors in the trust assessment. The other RC which contains the user ID is used to construct the BID and ensure the security of the framework. How the BID and RC pair are generated and used is a key component of our scheme. We elaborate more on the details in Section IV-C.

2. Contextual Provenance: The contextual provenance is a description of the sensing environment. It contains attributes such as sensing time, sensing location, sensor mode (i.e., whether it is a text report, voice clip, picture or video, etc), participants' traveling mode, and other optional contextual information. These contextual attributes usually have a big influence on the trust of the sensing reports.

According to a survey done by Christin et. al. [15], virtually all participatory sensing applications collect time and location information, thus underpinning the importance of these two factors. Other than time and location, we believe the type of sensor used and the participants' traveling mode also largely affect the reliability and correctness of the sensing data. For instance, usually we consider a picture or a video clip better than a text-only description. Also, inaccurate sensing reports tend to be generated when a participant is traveling at a very fast speed. Therefore, we make time, location, sensor mode and traveling mode as the four *contextual factors* that we are going to use for assessing the trust of the sensing reports.

B. Sensing Report Trust Assessment

When a report is received, the server first validates the anonymous RC in the user provenance by checking:

1) The RC has been signed by the server.
2) The RC is issued for the current task.

If the validation is passed, the server obtains the reputation level $\hat{R}(P_i)$ of the sender P_i. The server cannot associate $\hat{R}(P_i)$ with P_i because many participants could have the same reputation level. Though $\hat{R}(P_i)$ is not accurate, it gives the server a rough idea of how much the sender can be trusted.

The contextual provenance contains other factors that may affect the trust of the sensing report. Here, we only give a

general solution based on the provenance model we proposed, i.e., we assume all the four factors (time, location, sensor mode and traveling mode) affect the trust of a sensing report. It is absolutely possible that one or more of these factors need not to be considered or other contextual factors become important when we have a specific application. The system designer can easily tailor our scheme based on the application's needs.

A sensing report from a location faraway from the expected location is usually not as accurate as a report from a nearby location. We call the expected location and the actual location indicated in the contextual provenance the *target location* (denoted as L_t) and the *sensing location* (denoted as L_s) respectively. We denote $|L_s - L_t|$ as the distance between them. Spatial cloaking techniques may obfuscate the sensing location. In other words, the location provided in the contextual provenance might be a small area instead of an exact location point. We call this area as the *cloaking area* and denote D_c as its diameter. In this case, we use the central point of the cloaking area as the sensing location. We then formally define the *location distance factor* (denoted as Θ) as:

$$\Theta = e^{-D_c \cdot \alpha} \cdot (1 - e^{-|L_s - L_t| \cdot \alpha}) \tag{1}$$

where α is the *location sensitivity parameter* set by the system which controls the weight of the location factor's influence on the trust of sensing reports. The $1 - e^{-|L_s - L_t| \cdot \alpha}$ part of the equation makes Θ equal to 0 when $|L_s - L_t|$ equals to 0 and Θ approaches 1 when $|L_s - L_t|$ is large. The $e^{-D_c \cdot \alpha}$ part accounts for the uncertainty caused by the cloaking area. A maximum sensing distance and a maximum cloaking diameter can be set, so that if $|L_s - L_t|$ exceeds the maximum sensing distance or the reported D_c exceeds the maximum cloaking diameter, the sensing report will be discarded.

Time is another critical factor. Reports sensed at the expected time usually have the best quality. We call the expected time of the sensing task and the actual time contained in the contextual provenance the *target time* and the *sensing time*. We denote $|T_s - T_t|$ as the time gap between them. When temporal cloaking techniques are used, we call the resulting time interval as the *cloaking interval* and denote S_c as the length of the cloaking interval. Again, we use the middle point of the cloaking interval as the sensing time if time is cloaked. We define the *time gap factor* (denoted as Ω) as:

$$\Omega = e^{-S_c \cdot \beta} \cdot (1 - e^{-|T_s - T_t| \cdot \beta}) \tag{2}$$

where β is the *time sensitivity parameter* which controls the weight of the time factor's influence on the trust of sensing reports. Similar to the location factor, a maximum time gap and a maximum cloaking interval can be set.

The sensor mode and traveling mode are two other important factors that might affect the report quality, too. The system can define a weighting parameter for each sensor and traveling mode. As an example, Table I shows a list of system-defined sensor mode weighting parameters (denoted as λ) and traveling mode weighting parameters (denoted as μ).

We can calculate the *base trust* (denoted as $T_b(r)$) of the sensing report based on the reputation level and the four contextual factors as follows:

$$T_b(r) = \hat{R}(P_r) \cdot (1 - \Theta_r) \cdot (1 - \Omega_r) \cdot \lambda_r \cdot \mu_r \tag{3}$$

The base trust is merely a value we calculate based on the provenance. It is an important reference to us when a single report is received. However, in most cases, multiple sensing reports might be received for one sensing task. Different reports for the same task may be either mutually supportive or conflicting. Similar reports are considered supportive to each other, while conflicting reports compromise the trustworthiness of each other. Therefore, we can adjust trust based on the amount of supports and conflicts the reports get from each other. We group all the sensing reports for a particular sensing task in a collection C before the sensing task expires.

TABLE I: Sensor mode and traveling mode weighting parameters

Sensor Mode	$\boldsymbol{\lambda}$
Text	1.00
Voice	1.05
Picture	1.20
Video	1.30

Traveling Mode	$\boldsymbol{\mu}$
Standstill	1.0
Walking	0.98
Cycling	0.95
Driving @ $<$ 30 mph	0.94
Driving @ $>$ 30 mph	0.92

For data similarity measurement, there has been lots of work done in the field of data mining [16]. We assume any two sensing reports r and r' within a collection have a similarity score of $S(r, r')$ which ranges from -1 to 1, where -1 means completely conflicting and 1 means exactly the same. Now what we really care about is how to actually utilize the similarity scores to adjust the report trust. We assign a *similarity factor* Δ_r to sensing report r which belongs to a collection C_r as follows:

$$\Delta_r = \frac{\sum_{r, r' \in C_r, r \neq r'} S(r, r')}{|C_r| - 1} \cdot e^{-\frac{1}{|C_r|}} \cdot \gamma \tag{4}$$

where $|C_r|$ is the number of sensing reports in the collection C_r and γ is the *similarity weighting parameter* that controls the weight of the similarity adjustment. The rationale behind the term $e^{-\frac{1}{|C_r|}}$ is that the more reports are in the collection C_r, the better idea we would have about what is right and what is wrong. Thus, we increase the influence of the similarity factor as the number of report in a collection increases, but the rate of this increment should be slowed down and never exceed a threshold when the number of report becomes large.

Each sensing report is assigned with a similarity factor. A negative similarity factor means there are more conflicts in the collection and a positive similarity factor means there are more supports. Finally, we can obtain the *final trust* (denoted as $T_f(r)$) of the sensing report r as follows:

$$T_f(r) = T_b(r)(1 + \Delta_r) \tag{5}$$

Comparing the final trust $T_f(r)$ and the original reputation level $\hat{R}(P_r)$, it is easy for the server to generate a *reputation feedback level* f_R. Similar to the reputation level, f_R cannot be an accurate number, otherwise the server can associate the f_R with the original report later when f_R is being redeemed by the participant (more details in Section IV-C). Our suggestion is to predefine a number of discrete f_R levels based on the difference between $T_f(r)$ and $\hat{R}(P_r)$, and the number of f_R levels should not be too many in order to minimize the probability that the server can associate a f_R with its original report. There are many ways of doing so. A general guideline is, positive f_R should be given if $T_f(r) > \hat{R}(P_r)$, and vice versa. Also, negative feedbacks should affect the reputation more than positive feedbacks. This tallies with our intuition that a reputation can only be built up with a long time of consistent good behaviors, but a few bad incidences could ruin the reputation drastically. Table II gives an example solution.

TABLE II: Predefined reputation feedback levels

$T_f(r) - \hat{R}(P_r)$	f_R
(0.5, 1]	0.02
[0.1, 0.5]	0.01
[-0.1, 0.1]	0
[-0.5, -0.1)	0.025
[-1, 0.5)	0.05

TABLE III: List of notations

$A\|B$	Concatenation of message A and message B
K_{spub}	Public key of the server
K_{spriv}	Private key of the server
$\{M\}_{K_{spub}}$	Message M encrypted by K_{spub}
$[M]_{K_{spriv}}$	Message M signed by K_{spriv}

C. Anonymous Reputation Management

An Anonymous Reputation Management (ARM) scheme for participatory sensing applications needs to have the following attributes:

- A1 Sensing reports do not contain identity information and the server cannot associate a report with a particular participant by any means.
- A2 Multiple sensing reports from the same participant are not linkable.
- A3 A participant's reputation is determined by his/her past behaviors, and participants do not have control over the reputation update process.
- A4 Participants can demonstrate their reputation levels to the server without revealing their identities and they cannot lie about their reputation levels.

During a user registration, participants normally need to provide their personal information such as name, contact and payment information. Therefore, the user ID can be considered as the real-identity of a participant. To achieve A1, many anonymity schemes uses pseudonyms. Nevertheless, a stable pseudonym makes the reports from the same participant linkable and thus violates A2. If a participant does not change his/her pseudonym frequently enough, the real-identity could still be revealed by analyzing the location traces. A3 and A4 are challenging because the reputation is associated with the user ID in the reputation database and anonymity makes it hard to enforce the participants to follow the protocols. To solve these issues, our approach utilizes the Blind Signature technique [17] and make the report submission and reputation update as two separated processes. We illustrate the entire sensing task cycle in Figure 3. There are five crucial steps in this cycle, which are indicated as (1) - (5) in Figure 3. We now describe each of these steps in detail and the notations we use are listed in Table III.

1. Issue of Reputation Certificate (server side): First of all, when a participant P_i decides to take a sensing task, he/she needs to register with the server for this task before he/she sends out a sensing report. The participant does this by sending a *Task Registration Request* (TRR) which contains his/her user ID P_i and the corresponding Task ID TID. Task registration does not violate anonymity because the server would only know who wants to participate, but would not be able to link them with their actual sensing reports.

The server maintain a *task registration table*. When a TRR is received, the server registers the participant P_i for task TID by putting the tuple (P_i, TID) into the task registration table.

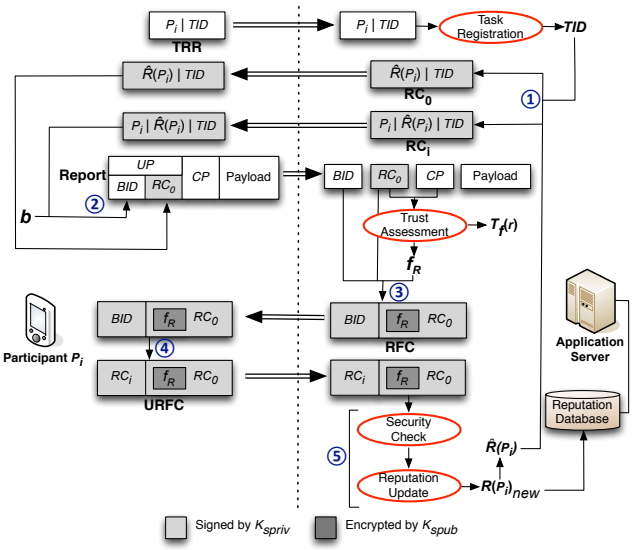


Fig. 3: An illustration of the anonymous report submission and reputation management in a sensing task cycle

After task registration, the server obtains P_i's reputation level $\hat{R}(P_i)$ based on his/her most recent reputation $R(P_i)$ (R_0 for new participants). A pair of RC's are created by the server, where one RC contains P_i (denoted as RC_i) and the other does not (denoted as RC_0).

$$RC_i = \left[P_i|\hat{R}(P_i)|TID\right]_{K_{spriv}} \tag{6}$$

$$RC_0 = \left[\hat{R}(P_i)|TID\right]_{K_{spriv}} \tag{7}$$

Both RC_i and RC_0 contain $\hat{R}(P_i)$ and TID and both of them are signed by the server. RC_0 is the anonymous RC that will be put in the user provenance by the participant, and RC_i is necessary for constructing the BID (explained in next step). Whenever a participant wants to participate in a new task, he/she has to obtain a refreshed RC pair for this specific task. TID is used to check if the RC_0 was issued for the current task when a sensing report is submitted.

2. Construction of Blinded ID (user side): As we described in SectionIV-A, every user provenance contains a *Blinded ID* (BID) of the sender. To construct the BID, the participant needs his/her RC_i and a random number b. b is chosen by the participant such that b is relatively prime to the server's public modulo N. Then, b is raised to the public exponent e modulo N, and the result $b^e \pmod N$ is used as a *blinding factor*. BID is the product of RC_i and the blinding factor:

$$BID \equiv RC_i \cdot b^e \pmod N \tag{8}$$

Every time a participant submits a report to the server, he/she can choose a different random number b, and thus making the BID different. Therefore, the BID cannot be used by the server to link reports from the same participant.

3.Generation of Reputation Feedback Coupon (server side): After assessing the trust of a sensing report, the server generates the reputation feedback level f_R for the sender (as described in Section IV-B). Then, a *Reputation Feedback Coupon* (RFC) is generated as follows:

$$RFC = \left[BID\right]_{K_{spriv}} \left|\left[\{f_R\}_{K_{spub}}|RC_0\right]_{K_{spriv}}\right. \tag{9}$$

where f_R is encrypted by the server's public key so that the participant cannot tell if it is a negative or positive feedback.

4. Ublinding RFC (user side): With the received RFC, the original report sender can obtain an *Unblinded RFC* (URFC) by removing the blinding factor based on the characteristics of blind signatures. The resulting URFC will be as follows:

$$URFC = \left[RC_i\right]_{K_{spriv}} \Big| \left[\left\{f_R\right\}_{K_{spub}} \Big| RC_0\right]_{K_{spriv}} \quad (10)$$

After getting the URFC, the participant chooses to wait a random period of time before the URFC is expired (if there is an expiration time), and then sends the URFC to the server to redeem it. The UFRC is signed by K_{spriv} so that no participant can forge a valid URFC at this stage.

5. Redemption of URFC (server side): When the server receives a URFC, a security check must be done on the URFC to make sure it passes the following requirements:

1) The private-key signatures and public-key encryptions are valid.
2) The two copies of $\hat{R}(P_i)$ and TID extracted from RC_i and RC_0 are consistent.
3) No URFC with the same P_i and TID has been redeemed before.
4) The URFC is not expired (optional).

If the URFC passes the validation, the server extracts P_i and f_R from the URFC and updates the corresponding entry in the reputation table. Now we can see that if an accurate value of f_R was used in a RFC, the server would be able to use it to associate P_i with the original sensing report.

V. Security Analysis

In this section, we will analyze and prove that the proposed ARM protocol can achieve our goals A1-A4 and the mechanism itself is secure.

Proposition 1. *The server cannot see the user ID from a sensing report.* (A1)

Every time a participant P_i sends a sensing report, the BID is included in the user provenance instead of the real user ID P_i. According to the characteristics of the Blind Signature technique, no information about P_i can be extracted from BID by the server.

Proposition 2. *The server cannot correlate the user ID with the original sensing report when URFC is redeemed.* (A1)

When a URFC is sent to the server for reputation redemption, the server can extract P_i, $\hat{R}(P_i)$, f_R and TID. P_i was blinded in BID and could not be seen by the server in the original sensing report. Based on the definition of $\hat{R}(P_i)$ and f_R, many different reports for the task TID would have the same $\hat{R}(P_i)$ and f_R. Thus, neither of them can be used by the server to correlate P_i with the original sensing report.

Proposition 3. *The server cannot link multiple reports sent from the same participant.* (A2)

A participant can choose a different blinding random number b for each sensing report he/she sends when BID is constructed. This makes BID for the same participant different for different sensing reports. The server cannot find any linkage between these BID's due to the randomness of b. RC_0 cannot be used to link reports from the same participant either, because RC_0 only contains $\hat{R}(P_i)$ and TID. Based on the definition of $\hat{R}(P_i)$, many different participants may have the same $\hat{R}(P_i)$ in their RC_0 for task TID.

Proposition 4. *A participant cannot redeem a URFC multiple times or redeem multiple URFC's for the same task without being detected.* (A3)

When the server receives a URFC for redemption, it extracts P_i and TID. If it has seen the same P_i and TID before, which indicates that either the participant is trying to redeem a URFC multiple times or the participant is trying to redeem multiple URFC's received from sending multiple reports for the same task. Both cases should be disallowed. If this happens, the participant is considered to have malicious intent and the server can apply a penalty on the participant's reputation.

Proposition 5. *A participant cannot redeem another collusive participant's URFC in order to get an unfair reputation update without being detected.* (A3)

According to how reputation feedback levels are given in our system, when two participants send the same good reports, the participant with lower reputation level tends to get a higher reputation feedback level. Two collusive participants may want to switch their URFC's for redemption in order to unfairly promote the reputation of the participant who already gained higher reputation. If two entire URFC's are switched and redeemed. The user ID in the RC_i can tell the server that the user is trying to redeem someone else's URFC. If only the $\left[\left\{f_R\right\}_{K_{spub}} \Big| RC_0\right]_{K_{spriv}}$ part of the two URFC's are switched, the inconsistency of $\hat{R}(P_i)$'s in RC_i and RC_0 will again warn the server about the malicious behavior.

Proposition 6. *A participant cannot refuse to redeem a URFC for participated tasks without being detected.* (A3)

An adversary who intentionally sends false data might refuse to redeem the URFC's because he/she knows most probably the feedback would be negative. A good participant who has obtained a high reputation might also never want to redeem any more URFC's to prevent his/her reputation from being decreased. Since the server has the task registration table, it can easily find out which registered participant(s) never redeemed a URFC for a particular task. To prevent this from happening, the server can choose to apply a reputation penalty higher than the worst negative feedback level.

Proposition 7. *The server can give both positive and negative reputation feedback to participants.* (A3)

First, the f_R in a RFC or URFC is encrypted by the server with its public key K_{spub}, a participant cannot decrypt $\left\{f_R\right\}_{K_{spub}}$ and see if f_R is a positive or negative feedback level. More importantly, according to Proposition 6, refusing to redeem a URFC will incur a bigger loss on the reputation than the worst negative feedback level.

Proposition 8. *A participant cannot forge a URFC or a RC without being detected.* (A3 & A4)

After a participant unblinds a RFC, the server's signature remains on the RC_i part and the $\{f_R\}_{K_{spub}}|RC_0$ part has its original signature from the server. Since only the server has the access to K_{spriv}, a participant cannot forge a URFC. A RC_i and RC_0 pair is also signed by K_{spriv} before they are issued to a participant, thus no participant can forge a RC.

Proposition 9. *A participant cannot demonstrate a higher reputation level in a sensing report with another collusive participant's RC without being detected.* (A4)

Since RC_0 does not contain P_i, it is possible for a participant to obtain another participant's RC_0 with a higher reputation level and use it in his/her own sensing report. Due to the anonymity, the server cannot detect it from the sensing

TABLE IV: Default parameter settings

Parameter	Value
Number of participants for each task	100
Number of adversaries in the participants	10
Nature of adversaries	0
Location sensitivity parameter α	0.2
Time sensitivity parameter β	0.2
Similarity weighting parameter γ	0.5
Maximum sensing distance	10
Maximum time gap	10
Maximum cloaking diameter	20
Maximum cloaking interval	20
Initial reputation R_0	0.5

report. However, when the participant redeems the URFC, the server compares RC_i and RC_0. Since RC_i contains P_i, it is impossible for a participant to use another participant's RC_i. Therefore, if a participant has used another participant's RC_0 with a higher reputation level, the $\hat{R}(P_i)$'s extracted from RC_i and RC_0 of the URFC will be inconsistent.

VI. Performance Evaluation

A. Simulation Setup

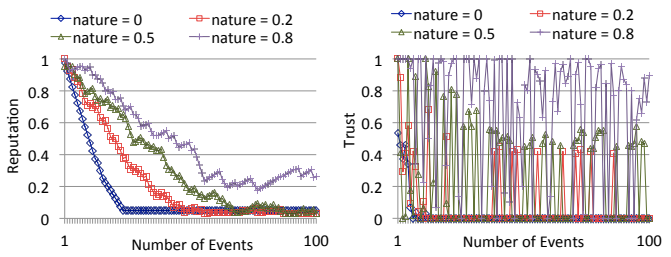
We implemented our scheme with Java simulation to measure the performance and accuracy of our trust assessment and reputation management. Since the communication links are not our concern, we implemented a server and multiple participants on a single Linux machine.

In our simulation tests, we define *good participant* as a participant that always sends correct sensing reports. However, an adversary does not necessarily always send false sensing reports. They may launch on-off attacks by sending correct reports in order to gain reputation and then only send false reports randomly or at a specific time. We define the *nature* of an adversary as the probability of the adversary sending correct reports. When an adversary sends a false report, we set the data to be completely opposite to the correct report and all the false reports support each other. In this case, we are looking at the worst case that all adversaries collusively send data to cause the biggest possible disturbance to the system.

Table IV lists our default parameter settings. For sensor and traveling mode weighting parameters and reputation feedback levels, we use Table I and Table II as our default settings. When each participant sends a sensing report, we generate a random sensing location and sensing time within the maximum sensing distance and maximum time gap. The cloaking area diameter and cloaking time interval are also randomly generated within the maximum values. The sensing mode and traveling mode are also randomly selected in Table I.

B. False Positive and False Negative Rates

First of all, to measure the accuracy of our sensing report trust assessment, we carried out a series of tests to see the false positive (FP) and false negative (FN) rates of our trust assessment with our default settings. FP means a report is actually correct but the calculated trust is lower than an *alarm threshold*. On the contrary, FN means the calculated trust for a false report is higher than the alarm threshold. The alarm threshold is a trust level below which we will consider the sensing report untrustworthy. It can be set based on the needs of the specific application. We tested FP and FN rates for



(a) Reputation of a particular adversary with varying nature

(b) Trust of sensing reports from a particular adversary with varying nature

Fig. 4: Impact of an adversary's nature on reputation and trust

reports received from a participant with different nature for various alarm thresholds and the results are shown in Table V. Each of these values is a result based on testing 10000 sensing reports. In the table, (x) means the alarm threshold is x. We can see the overall FP and FN rates are very low (approximately 0 when the alarm threshold is set to be 0.5). The FP and FN rates increase for more strict alarm thresholds (i.e., FP with a higher alarm threshold or FN with a lower alarm threshold). However, we can see FN rate is still close to 0 even when the alarm threshold is 0.2. That means, when a sensing report is false, there is a very minimal probability that its trust value is going to be higher than 0.2. FP rates are generally higher than its counterpart FN rates, due to the randomness introduced by the contextual factors in our simulation, but definitely within an acceptable range.

Table V only shows the false positive and false negative rates under the default parameter settings. One can imagine that when the system settings change, our calculated trust and reputation would change, too. In the rest of this section, we will show how some important system parameters would affect trust and reputation. In each test, we vary certain parameters to see their impacts, and we will specify these parameters. For other parameters we do not specifically mention, they are set as the default values.

C. Impact of Adversary's Nature

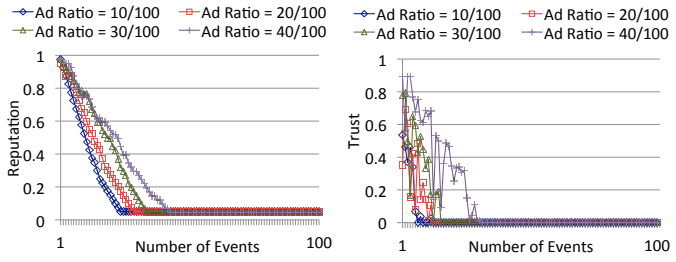
First, we want to see how an adversary's nature would influence his/her reputation and his/her reports' trust. We have four adversaries with a nature value of 0, 0.2, 0.5 and 0.8 respectively. To test the worst case, we assume all of them have gained a reputation value of 1 before the test. A total number of 100 tasks for this test were run.

Figure 4 (a) shows how the reputation of an adversary changes as the number of tasks increases. When an adversary has a nature of 0 (i.e., always reports false data), his/her reputation drops down very quickly until a level very close to 0. An adversary who randomly sends correct data (with nature 0.2, 0.5 and 0.8) can slow down this dropping process. However, eventually the reputation still drops down to a very low level even if false data are sent with a small probability (the *0.8-nature* curve). This is because negative feedback levels have larger influence on the reputation. We set both the reputation feedback levels to be relatively small in order to prevent that one single task affects the reputation too much.

Next, we examine the computed trust values of the sensing reports sent by adversaries for the same test settings. Figure 4 (b) shows the result. The *0-nature* curve indicates that reports

TABLE V: False positive rates and false negative rates with default settings

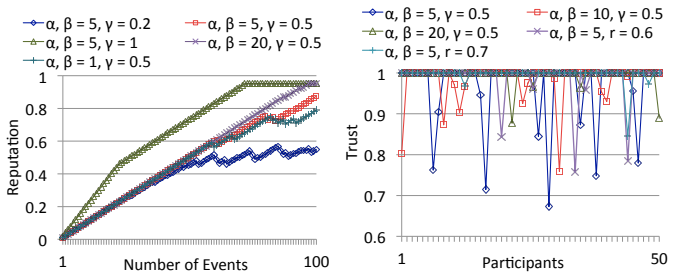
Nature	FP (0.5)	FN (0.5)	FP (0.6)	FN (0.4)	FP (0.7)	FN (0.3)	FP (0.8)	FN (0.2)
1 (good participant)	~ 0	N.A.	0.31%	N.A.	1.82%	N.A.	4.49%	N.A.
0.8	~ 0	~ 0	0.34%	~ 0	1.86%	~ 0	4.68%	~ 0
0.5	0.02%	~ 0	0.71%	~ 0	2.52%	~ 0	5.72%	0.01%
0.2	0.05%	~ 0	1.01%	~ 0	2.95%	~ 0	7.11%	0.12%
0	N.A.	~ 0	N.A.	~ 0	N.A.	~ 0	N.A.	0.23%



(a) Reputation of a particular adversary with varying adversary ratio

(b) Trust of sensing reports from a particular adversary with varying adversary ratio

Fig. 5: Impact of adversary ratio on reputation and trust



(a) Reputation of a good participant with varying α, β and γ

(b) Trust of sensing reports from 50 good participants with varying α, β and γ

Fig. 6: Impact of α, β and γ on reputation and trust

from an adversary with nature of 0 have a non-zero trust at the beginning when the reputation is still high, and the curve stays at 0 after a couple of tasks. The trust of reports from adversaries with nature 0.2, 0.5 and 0.8 fluctuates because of the mixture of correct and false reports. As expected, we observe that the higher nature an adversary has, the higher probability that his/her reports will get a high trust value. Take the *0.8-nature* curve as an example, since the adversary sends correct report 80% of the time, most of the values on the curve lies on the high trust range (0.6 to 1). We capture the 20% false reports because most of them have a trust value of 0.

D. Impact of Adversary Ratio

In our next test, we set the nature of all adversaries to be 0, which is the worst case and we vary the ratio of adversaries in the network. We set the number of adversaries as 10, 20, 30 and 40 respectively, out of 100 participants.

The result for the reputation updates is shown in Figure 5 (a). It is clear that as the ratio of adversaries increases, the reputation for a particular adversary drops down more slowly. This is because we made the adversaries to be colluding and their reports gain more supports from each other. However, as long as the good participants are more than the adversaries in the network, an adversary's report will get a negative reputation feedback with a high probability. This is why even when 40 out of 100 participants are colluding, their reputation still keeps decreasing until reaching a level close to 0.

Again, for the same settings, we test the trust assessment and our result is shown in Figure 5 (b). The curves follow the similar trend as Figure 5 (a). However, the trust curves fluctuate much more than the reputation curves, which is the expected result. The reason is the contextual factors and the similarity factor affect the trust of an individual report much more than they would affect the overall reputation.

E. Impact of α, β and γ

The location sensitivity parameter α, time sensitivity parameter β and similarity weighting parameter γ are crucial to our framework. Therefore, we want to investigate how these parameters affect trust and reputation. α and β could decrease the trust of a sensing report because of unideal sensing location and time. γ could increase and decrease the trust of a sensing report depending on the amount of support and/or conflict it gets from other reports. The reputation of a good participant and the trust of his/her sensing reports could better demonstrate the effects varying α, β and γ. Hence, we look at the reputation of a good participant and the trust of good sensing reports. Furthermore, since α and β work in a very similar way, we vary them together to see their impacts.

Again, to test the worst case, we assume the good participant has an initial reputation of 0. As shown in Figure 6 (a), the reputation of the good participant increases at different rates when α, β and γ is set as different values. When γ is large (the α, β = 0.2, γ =1 curve) or when α and β is small (the α, β = 0.05, γ = 0.5 curve), the report similarity overwhelms the randomness in the contextual factors and therefore the good participant always gets positive feedback. When α, β becomes larger or γ becomes smaller, the randomness of the contextual factors starts to appear. Hence, a portion of the sensing reports may get negative feedback due to the negative impacts from the contextual factors even though the participant is good. Therefore, we can see that the reputation of a good participant goes up and down on some curves. It does not mean these cases are bad. If the application is sensitive to the context, it is expected that reports with unideal contextual factors decrease the senders' reputation.

To show the impacts of α, β and γ on individual sensing reports clearly, we only look at one task and we let 50 different good participants that have a reputation of one with random sensing locations and times send their sensing reports. Figure 6 (b) shows how α, β and γ affect the trust of these sensing reports. It is clear that the randomness of the location and time factors could give the reports different trust values and large α and β magnify the impacts of these two factors. When γ is large, the report similarity have bigger influence

on the trust. Since all the reports are good and have the same amount of supports from each other, bigger influence from report similarity makes the randomness of location and time less prominent. Therefore, based on the time and location sensitivity of the system, proper α and β values should be chosen. Based on the choice of α and β and other system characteristics (e.g. the amount of sensing reports for each task), a proper γ value needs to be set in order to prevent the similarity factor from having too little or too much influence.

VII. Discussion and Future Work

Our solution is depending upon a redundant number of participants. Data has shown that applications like Gigwalk is undergoing a big growth [18]. We believe participatory sensing applications with huge user bases are soon going to be emerged. For a system with a large user base, assuming most of the users are good participants, there should be redundant number of users with reputation levels from average to high. Therefore, we argue that good participants' anonymity can be well protected by using our approach.

Most of reputation systems are vulnerable to Sybil attacks, i.e., an attacker obtains multiple identities. As an approach to mitigate Sybil attacks, the user registration process can enforce people to provide some information so that they cannot freely register unlimited number of accounts. For example, the device IMEI number might be requested for registration so that each mobile device can maximally register for one account. Since users do not interact with each other in participatory sensing, Sybil accounts cannot promote each other's reputation as in traditional reputation systems like eBay. The main incentive for Sybil attacks now becomes sending false data collusively to disrupt the trust and reputation calculation. We have shown that our system is collusion-resilient if the number of good reports exceeds the number of false reports.

Another possible attack scenario is that a single adversary sends multiple reports for a specific task using the same RC_0 obtained from the task registration, and only redeem one of the URFC's received for those reports. The trust assessment may be biased by such an attack. However, in our scheme, the server knows how many reports are supposed to be received based on the task registration table. It is easy for the server to detect it when such an attack happens. When the exceeded amount of reports are larger than a certain threshold, the server can choose to discard all the reports and re-distribute the task.

In the future, we will look into the possibility of detecting the source of attacks while maintaining anonymity for good participants. Anonymous blacklisting techniques [19] are good directions to explore. In addition, we will deploy our scheme into a real participatory sensing application and carry out more in-depth evaluations on the security and privacy protection.

VIII. Conclusion

Trust and anonymity are two conflicting objectives in a participatory sensing application. In this work, we proposed the ARTSense framework to achieve both of them at the same time. First, we proposed a novel provenance model which serves as the basis of our trust assessment for the sensing reports. To achieve anonymity, our ARM protocol separates the data reporting process and reputation update process. No user identity information is revealed in each individual sensing report, and furthermore, the server cannot associate multiple reports from the same participant because the usage of Blind ID. Our reputation feedback and redemption process enforces measuring user reputation without violating anonymity and it allows both positive and negative reputation feedback. Our entire framework is proven to be able to achieve the pre-defined anonymity and security requirements, and resilient to malicious behaviors such as newcomer, on-off and collusion attacks. Our simulation results confirmed that with proper choices for the system parameters, different participatory sensing applications can be accommodated, and both user reputation and data trust can be accurately captured.

IX. Acknowledgments

This work was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official polices, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy right notation here on.

References

[1] "Gigwalk." http://www.gigwalk.com.
[2] "mCrowd." http://crowd.cs.umass.edu/.
[3] I. Krontiris, F. Freiling, and T. Dimitriou, "Location privacy in urban sensing networks: research challenges and directions," *IEEE Wireless Communications*, vol. 17, no. 5, pp. 30–35, 2010.
[4] A. Kapadia, D. Kotz, and N. Triandopoulos, "Opportunistic sensing: Security challenges for the new paradigm," in *IEEE International Communication Systems and Networks and Workshops.*, 2009.
[5] P. Kalnis, G. Ghinita, K. Mouratidis, and D. Papadias, "Preventing location-based identity inference in anonymous spatial queries," *IEEE Transactions on Knowledge and Data Engineering*, 2007.
[6] B. Gedik and L. Liu, "Protecting location privacy with personalized k-anonymity: Architecture and algorithms," *IEEE Transactions on Mobile Computing*, vol. 7, pp. 1–18, 2008.
[7] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal on Uncertainty Fuzziness and Knowledgebased Systems*, vol. 10, no. 5, pp. 557–570, 2002.
[8] K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Participatory privacy in urban sensing," in *MODUS Workshop*, 2008.
[9] M. Shin, C. Cornelius, D. Peebles, A. Kapadia, D. Kotz, and N. Triandopoulos, "Anonysense: A system for anonymous opportunistic sensing," *Pervasive and Mobile Computing*, 2010.
[10] E. De Cristofaro and C. Soriente, "Pepsi: Privacy-enhanced participatory sensing infrastructure," in *ACM WiSec*, 2011.
[11] A. Dua, N. Bulusu, W. Feng, and W. Hu, "Towards trustworthy participatory sensing," in *Proceedings of the 4th USENIX Conference on Hot Topics in Security*, pp. 8–8, USENIX Association, 2009.
[12] X. Wang, K. Govindan, and P. Mohapatra, "Collusion-resilient quality of information evaluation based on information provenance," in *IEEE 8th Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2011.
[13] W. Müller, H. Plötz, J.-P. Redlich, and T. Shiraki, "Sybil proof anonymous reputation management," in *ACM International Conference on Security and Privacy in Communication Networks*, 2008.
[14] E. Androulaki, S. G. Choi, S. M. Bellovin, and T. Malkin, *Reputation systems for anonymous networks*. 2008.
[15] D. Christin, A. Reinhardt, S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *Journal of Systems and Software*, 2011.
[16] C. Beecks, M. Uysal, and T. Seidl, "A comparative study of similarity measures for content-based multimedia retrieval," in *IEEE International Conference on Multimedia and Expo (ICME)*, 2010.
[17] D. Chaum, "Blind signatures for untraceable payments," in *International Crytology Conference*, pp. 199–203, 1982.
[18] "Gigwalk - the future of your work is mobile." http://www.gigwalk.com/future.
[19] R. Henry and I. Goldberg, "A survey of anonymous blacklisting systems," *Technical Report, Centre for Applied Cryptographic Research, University of Waterloo*, 2010.