Enabling Reputation and Trust in Privacy-Preserving Mobile Sensing

Xinlei (Oscar) Wang, *Student Member, IEEE,* Wei Cheng, *Member, IEEE,*
Prasant Mohapatra, *Fellow, IEEE,* and Tarek Abdelzaher, *Member, IEEE*

Abstract—Mobile sensing is becoming a popular paradigm to collect information from and outsource tasks to mobile users. These applications deal with lot of personal information, e.g., identity and location. Therefore, we need to pay a deeper attention to privacy and anonymity. However, the knowledge of the data source is desired to evaluate the trustworthiness of the sensing data. Anonymity and trust become two conflicting objectives in mobile sensing. In this paper, we propose *ARTSense*, a framework to solve the problem of "trust without identity" in mobile sensing. Our solution consists of a privacy-preserving provenance model, a data trust assessment scheme and an anonymous reputation management protocol. In contrast to other recent solutions, our scheme does not require a trusted third party and both positive and negative reputation updates can be enforced. In the trust assessment, we consider contextual factors that dynamically affects the trustworthiness of the sensing data as well as the mutual support and conflict among data from difference sources. Security analysis shows that ARTSense achieves our desired anonymity and security goals. Our prototype implementation on Android demonstrates that ARTSense incurs minimal computation overhead on mobile devices, and simulation results justify that ARTSense captures the trust of information and reputation of participants accurately.

Index Terms—Mobile Sensing, Location Privacy, Anonymity, Data Trust, Reputation.

1 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 mobile sensing model, also known as participatory sensing and urban sensing, 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 mobile 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 use the general public to capture geo-tagged images, videos, audio snippets, or fill out surveys. Mobile users 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 3D photosynthesis of businesses and restaurants in Bing Map. Moreover, a notable number of other mobile sensing applications have also emerged for collecting more specific information such as traffic [3], [4], noise pollution [5], cyclist experiences [6], and consumer pricing information [7].

Sharing sensed data tagged with spatio-temporal information could reveal a lot of personal information, such as a user's identity, personal activities, political views, health status, etc. [8], which poses threats to the participating users. Therefore, mobile sensing requires a deeper attention to privacy and anonymity, and a mechanism to preserve users' location privacy and anonymity is mandatory. Another dimension of data security in mobile sensing is the reliability of the sensed data. In mobile 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 mobile sensing system provides full anonymity, 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 [9].

There have been plenty of research efforts that have investigated privacy techniques for anonymous data collection in location based services (LBS) and particularly in mobile sensing systems. Most of other 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 mobile sensing networks. Compared with a few other existing solutions to the similar problem, our scheme does not require a trusted third party and both positive and negative reputation updates can be enforced while maintaining the

- *X. Wang and P. Mohapatra are with the Department of Computer Science, University of California, Davis, CA, 95616.*
  E-mail: {xlwang, pmohapatra}@ucdavis.edu.
- *W. Cheng is with the Department of Computer Science, Virginia Commonwealth University, VA, 23284.*
  E-mail: wcheng3@vcu.edu.
- *T. Abdelzaher is with the Department of Computer Science, University of Illinois at Urbana Champaign, IL, 61801.*
  E-mail: zaher@cs.uicu.edu.

desired user anonymity. In addition, we do not perform our trust assessment based on only user reputations but also other contextual factors that may dynamically affect the trustworthiness of the sensing data as well as the level of mutual support and conflict among sensing data received from difference sources. We also introduce a report flooding attack that has never been discussed in the context of mobile sensing, and proposed a way to utilize an anonymous blacklisting technique to defend against such attacks.

To summarize, the **contributions** of our work include:

1) A novel provenance model for mobile 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, privacy-preserving contexts such as location, time and other contextual factors, as well as mutual support and conflict among multiple sensing data.
3) An anonymous reputation management mechanism is presented to maintain the anonymity properties while also enforce positive or negative user reputation updates.
4) The Report Flooding attack is introduced and how an anonymous blacklisting scheme can be used in our scheme is discussed to defend again such attacks.

The rest of the paper is organized as follows. We highlight the related work of data security in mobile sensing in Section 2. In Section 3, 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 4. The security analysis of our scheme is given in Section 5 and performance evaluations based on prototype implementation and simulation experiments are presented in Section 6. We give a discussion of several additional privacy and security concerns in Section 7. Finally, Section 8 concludes the paper and talks about our future work.

2 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 [10], [11], where a 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 [12], 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 [13]–[16] have specifically studied the privacy in mobile sensing. In [13], the concept of participatory privacy regulation is introduced. In [14]–[16], different approaches are proposed, which focus on how participants upload the collected data to the server without revealing their identities. Most of them are based on cloaking techniques for protecting the location privacy of participants.

There have been numerous trust systems proposed toward the data reliability in mobile ad hoc networks, traditional wireless sensor networks as well as mobile sensing networks, for example, [17]–[19]. These approaches mainly focus on how the trustworthiness of the data shared in the network can be evaluated and how the reputation of the network entities which process the data can be maintained. For mobile sensing applications in particular, Huang et al [20] proposed a reputation system that employs the Gompertz function for computing device reputation score as a reflection of the trustworthiness of the contributed data. None of these solutions considered the high requirement for privacy and anonymity in the context of mobile sensing.

More recently, several privacy-aware reputation schemes [21]–[23] have also been proposed in the context of mobile sensing. In [21], the authors presented a scheme that utilizes multiple pseudonyms for each user and reputation values are transferred between different pseudonyms that belong to the same user. This scheme requires a trusted server to handle the reputation transfers between multiple pseudonyms of the same user and maintain the mappings between the real user identity and their pseudonyms. In [22], the authors proposed a similar solution named IncogniSense, which generates periodic pseudonyms by utilizing blind signatures and dynamically cloaks exact reputation values into reputation groups. It eliminates the assumption that the reputation and pseudonym manager must be trusted. However, as a separate party in the system, it still incurs additional management overhead. Another solution based on blind signature techniques was proposed by Li et al [23]. The authors looked at the problem from an incentive point of view, aiming to allow mobile users to earn credits by contributing data without leaking which data they have contributed. Therefore, they do not consider the necessity of penalizing malicious users in their privacy-aware incentive model.

In contrast to the existing solutions, our system does not require a trusted third party and both positive and negative reputation updates can be enforced while maintaining the appropriate level of user anonymity. In addition, we developed a novel provenance model for mobile sensing applications which serves as the basis of our sensing data trust assessment. From the trust analysis perspective, our system is different from the above mentioned schemes in that we do not only base our trust assessment on user reputation values but also other dynamic contextual factors that may affect the trustworthiness of the sensing data as well as the mutual support and conflict among sensing data received from difference sources. A preliminary result of this effort was

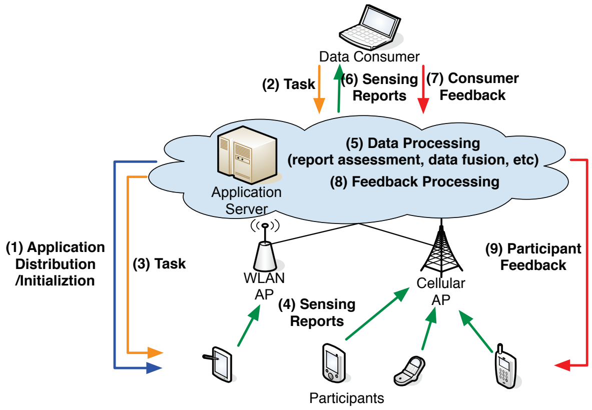


Fig. 1. Architecture of a mobile sensing system

presented in [24].

3 Problem Formation

3.1 System Architecture

Different mobile sensing applications may have different system models. To make it more specific, we consider a typical mobile 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). Some initialization work, e.g., user registration and privacy settings, should be done at this stage too.

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, 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.

In such a mobile sensing network, the network identifiers, e.g., IP addresses, could reveal the identities or locations of the participants [25]. 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.

3.2 Definitions of Trust and Reputation

A crucial part of the system is the assessment of the reliability and correctness of the sensing data reported by the participants. 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 [19] 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". Before diving into the details of our scheme, we first give formal definitions for these terms.

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.

3.3 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 functionalities. Malicious personnel behind the server may exploit the data collected by the application server but no one should be able to control how the application server performs its jobs, which includes user registration, key management, issuing credentials, task distribution, trust assessment and reputation management. As we described in Section 3.1, 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 [8]. Thus, unlinkability of the sensing reports sent by a single participant is an important desired security property of our solution.

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 tasks (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.

Since securing provenance is not the focus of this work, we assume provenance information is generated by a trusted middleware and the transmission of provenance is protected by a provenance security technique [26]. 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.

4 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 mobile sensing. The entire framework consists of three components: *provenance model*, *sensing report trust assessment* and *anonymous reputation management*. We present each of these components in detail in this section.

4.1 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.

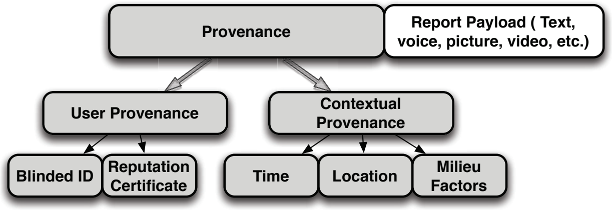


Fig. 2. Structure of a sensing report

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 an 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 4.3.

2. Contextual Provenance: The contextual provenance is a description of the sensing environment. It contains attributes such as sensing time, sensing location, 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. [27], virtually all mobile sensing applications collect time and location information, thus underpinning the importance of these two factors. One thing to be noted is, time and location are also the two factors that are closely associated with participants' privacy. Since we assume spatial/temporal cloaking techniques are used on each individual sensing report, these two factors in the contextual provenance may not contain precise information.

In addition to time and location, we believe that other contextual factors may also largely affect the reliability and correctness of the sensing data. These contextual factors could be the type of data, type of mobile device, battery level of the mobile device, participant's traveling speed, weather condition, etc. For instance, we consider a picture or a video clip better than a text-only description. Inaccurate sensing reports tend to be generated when a participant is using an older version of a mobile device or traveling at a very fast speed. The level of influence of such factors are very specific to the actual application scenario. We define such factors as the *milieu factors*. Mobile sensing applications may require different milieu factors to be included in the contextual provenance as property-value annotations.

4.2 Sensing Report Trust Assessment

ARM is an important component of the entire mobile sensing system, it provides a foundation to achieve our ultimate goal of the this paper, i.e., trust assessment for the sensing reports. Before talking about the details of our privacy preserving reputation management mechanism. We first describe our approach to assess the trust of sensing reports.

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.

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:

TABLE 1
Sensor mode and traveling mode weighting parameters

Data Type	$\boldsymbol{\lambda_{dt}}$
Text	1.00
Voice	1.05
Picture	1.20
Video	1.30
Traveling Mode	$\boldsymbol{\lambda_{tm}}$
Standstill	1.0
Walking	0.98
Cycling	0.95
Driving @ $<$ 30 mph	0.94
Driving @ $>$ 30 mph	0.92

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

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

In addition to the location and time, other milieu factors in the contextual provenance could be highly important and might affect the report quality, too. However, how much a particular milieu factor affects the report quality is really specific to the sensing task. There is not a universal way to compute the importance level of a particular milieu factor for different mobile sensing applications or under different circumstances for the same application. Without loss of generality, we leave the milieu factor selection approach open to the actual system designers. We suggest system designers to carefully select the milieu factors to be required in the contextual provenance and define a weight (denoted as λ) for each possible alternative of a milieu factor value. As an illustrating example, Table 1 shows a list of weights for different data types (λ_{dt}) and traveling modes (λ_{tm}). When a sensing report is received, the server is able to calculate a *synthesized milieu factor weight* (denoted as Λ) based on λ of each milieu factor. A simple way to do so is to get the product of all λ's.

We can calculate the *base trust* (denoted as $T_b(r)$) of the sensing report based on the reputation level and the synthesized milieu factor weight as follows:

$$T_b(r) = \min\{\hat{R}(P_r) \cdot (1 - \Theta_r) \cdot (1 - \Omega_r) \cdot \Lambda_r, 1\} \tag{3}$$

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.

For data similarity measurement, there has been lots of work done in the field of data mining [28]. 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 4.3). 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 2 gives an example solution.

TABLE 2
Predefined reputation feedback levels

$\boldsymbol{T_f(r) - \hat{R}(P_r)}$	$\boldsymbol{f_R}$
(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 3
List of notations

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

4.3 Anonymous Reputation Management

An Anonymous Reputation Management (ARM) scheme for mobile 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 [29] 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 3.

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.

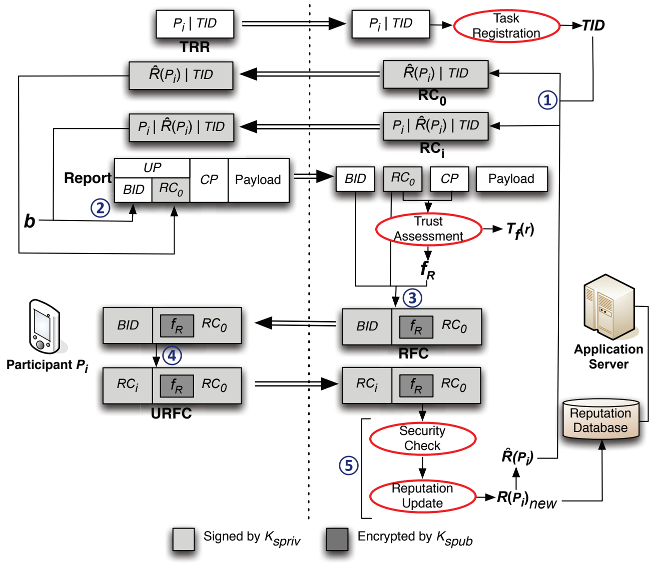


Fig. 3. An illustration of the anonymous report submission and reputation management in a sensing task cycle

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. 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 RCs are created by the server, where one RC contains P_i (denoted as RC_i) and the other does not (denoted as RC_0).

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

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

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 Section4.1, 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 (mod N) is used as a *blinding factor*. BID is the product of RC_i and the blinding factor:

$$BID \equiv RC_i \cdot b^e (\text{mod } N) \tag{8}$$

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 4.2). Then, a *Reputation Feedback Coupon* (RFC) is generated as follows:

$$RFC = \left[BID\right]_{K_{ss}} \Big| \Big| \left[\{f_R\}_{K_{sp}} |RC_0\right]_{K_{ss}} \tag{9}$$

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:

$$URFC = \left[RC_i\right]_{K_{ss}} \Big| \Big| \left[\{f_R\}_{K_{sp}} |RC_0\right]_{K_{ss}} \tag{10}$$

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_{ss} 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 an RFC, the server would be able to use it to associate P_i with the original sensing report.

4.4 Report Flooding

A subtle attack that our above construction is susceptible to is when a single adversary sends more than one reports for a specific task using the same RC_0 obtained from the task registration, and only redeem one of the URFCs received for those reports. We call this kind of attacks the *Report Flooding* attack.

The trust assessment could be biased by a report flooding attack. However, since the server knows how many reports are supposed to be received based on the task registration table, it is able to detect such attacks when the number of received reports exceeds the registered task applicants. The server can then choose to discard the reports and re-distribute the task when the exceeded amount of reports is larger than a certain threshold. Therefore, attackers cannot gain unfair reputation from doing so.

Due to anonymity, when such an attack happens, the server is not be able to tell who the attacker is. If an attacker launches such an attack every time a particular task is re-distributed, it becomes a DoS attack. In the rest of this section, we would like to discuss the possibility of detecting the source of such attacks with an *anonymous blacklisting* technique and how an anonymous blacklisting scheme can be plugged into ARTSense. Considering the fact that the computing resources in a mobile computing environment is often limited and an anonymous blacklisting technique usually introduces high overhead to the system, it is up to the system designer to decide if this additional functionality is necessary.

An anonymous blacklisting scheme works in a way that users (participants) authenticate themselves anonymously with the server, the server is able to revoke access from any users that misbehave without knowing their real identities or credentials. There are a number of anonymous blacklisting schemes in literature [30]. A particular scheme, BLAC [31], eliminates the requirement of a trusted third party, which makes it a better choice for our application scenario over the other schemes.

In BLAC, a *ticket* is presented by a user to the server during each execution of the authentication protocol, in order to prove that he/she is a legitimate user and he/she is not on the blacklist maintained by the server. A ticket is an output of an non-invertible mapping of the user's unique credentials. The tickets from the same user are unlinkable by taking as input some randomness so that the server cannot tell if two authentications are from the same user. More importantly, based on non-interactive *Signature Proof of Knowledge* protocols, a ticket is made to be provable for its correctness. That is, whether a ticket belong to the claiming user can be verified by the server without knowing the identity/credential of the user. A blacklist is a list of tickets for which the users are judged as having misbehaved by the server during the authenticated session. Due to space limitations, we omit the construction details of BLAC.

To plug BLAC into ARTSense and protest against report flooding, a unique credential should be given to each participant who registers for a particularity task in addition to RC_0 and RC_1. This credential is equivalent to the private credential issued in the registration protocol of BLAC. Before submitting a report to the server, a participant needs to run the anonymous authentication protocol of BLAC and prove to the server that he/she is not on the blacklist. Notice that the blacklist now is not for misbehaving users. Instead, each task should have a separated instance of blacklist which maintains the tickets of those users who have submitted a report for this particular task. In this case, once a participant submitted a report to the server, the server immediately adds his/her authentication ticket to the blacklist for that particular task. The security properties of the BLAC authentication protocol assures that a participant cannot authenticate himself/herself successfully more than once and thus submit more than one report for the same task.

BLAC is often criticized due to the fact that it scales linearly in the size of the blacklist [30]. It becomes impractical for many real-world applications because a blacklist of a thousand users makes it take several seconds to get a user authenticated. For large service providers with millions of users, the performance of BLAC is unacceptable. However, it would be rather rare that a single mobile sensing task requires that many participants. Therefore, each blacklist in our case would be only a short list of the participants most of the time. This makes the performance of BLAC acceptable with the way we utilize it.

5 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 BIDs 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 URFCs 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 URFCs 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 URFCs for redemption in order to unfairly promote the reputation of the participant who already gained higher reputation. If two entire URFCs 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 $\left[\{f_R\}_{K_{sp}}|RC_0\right]_{K_{ss}}$ part of the two URFCs 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 URFCs 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 URFCs 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)

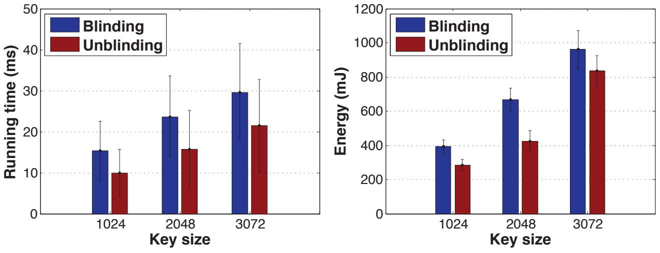
First, the f_R in an RFC or URFC is encrypted by the server with its public key K_{sp}, a participant cannot decrypt $\{f_R\}_{K_{sp}}$ and see if f_R is a positive or negative feedback level. More importantly, according to Proposition 6, refusing to redeem a URFC will incur a bigger loss on the reputation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

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

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



(a) Running time of the blinding and unblinding phases

(b) Energy consumption of the blinding and unblinding phases

Fig. 4. Computational measurement of mobile clients

6 Performance Evaluation

6.1 Prototype Implementation

We implemented a prototype client application on Android with Java to test the client side's performance, since the computational resources needed is critical for a mobile application. For each complete task cycle, there are two major phases a mobile client needs to execute: (1) sending a sensing report and (2) processing a RFC for later redemption. The blinding and unblinding are the two cryptographic operations that consume most of the computational resources on the client side for these two phases receptively. Therefore, we aim to measure the running time and energy consumption for a client application to construct a blinded ID (step ② in Figure 3) and to unblind a RFC (step ④ in Figure 3). Our experiments are carried out on a Samsung Nexus S device equipped with 1GHz processor, 512MB RAM, and running Android OS 4.1.1. The PowerTutor tool [32] is used to obtain the energy consumption measurement results. We use RSA blind signature [33] in our implementation. To study the impact of RSA key size on the performance of our client application, we test three key size settings representing three different security levels: 1024-bit, 2048-bit, and 3072-bit.

Figure 4 shows the results of our experiments. The results are based on 500 runs of the blinding and unblinding phases on the mobile device.

It is expected that the computational time and energy consumption increases with the key size since larger key size introduce more complexity to the blinding and unblinding computations. Under the three different key settings, the average computational time never exceeds

30 ms for both blinding and unblinding phases, which is considered very low. However, the energy consumption of both phases is fairly low. Based on our testing result, a fully charged battery for a Samsung Nexus S phone (1500mAh) can support over ten thousand blinding plus unblinding operations when the 2048 bit key size is used.

There are two reasons that the blinding phase requires more computational resources than the counterpart unblinding phase: (1) the modular exponentiation involved in the blinding computation makes it slightly more expensive than the unblinding computation; (2) the input data of the blinding computation (i.e., RC_i) is larger than the input data of the unblinding computation (i.e., BID).

6.2 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 the server and 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 5 lists our default parameter 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. It should be noted that these maximum threshold values are to be pre-defined based on the specific needs of the actual application. For example, a traffic sensing application may require the maximum sensing distance to be a hundred meters while a noise pollution sensing application may loosen such requirement. It is actually the location and time sensitivity parameters that determines the trust scores when the these maximum threshold values are set. Therefore, the location and time sensitivity parameters (α and β) must be adjusted accordingly in order for the resulting trust scores to be in a reasonable range. The maximum sensing distance and maximum time gap in Table 5 are set for the ease of calculation and the location and time sensitivity parameters are then adjusted and selected. The maximum cloaking factor (mcf) determines the size of the maximum cloaking diameter and maximum cloaking interval with respect to the maximum sensing distance and maximum time gap. A mcf of 2 in Table 5 means we allow each participant to cloak his/her location (time) to be in a cloaking area (cloaking interval) whose diameter (length) is maximumly two times of the maximum sensing distance (maximum time gap). The similarity weighting parameter (γ) controls the influence of the conflict and support level getting from other sensing reports. Similar to α and β, γ must be carefully chosen based on the other dynamics in the actual mobile sensing application in order for the resulting trust scores to be in a reasonable range. The impact of the choices of α, β and γ is evaluated and presented in Section 6.6.

TABLE 5
Default parameter settings

Parameter	Value
Number of participants for each task	100
Number of adversaries in the participants	10
Nature of adversaries	0
Initial reputation (R_0)	0.5
Maximum sensing distance	10
Maximum time gap	10
Maximum cloaking factor (mcf)	2
Location sensitivity parameter (α)	0.2
Time sensitivity parameter (β)	0.2
Similarity weighting parameter (γ)	0.5

In addition to the listed parameter settings, we generate a random synthesized milieu factor weight (Λ) in the range of $0.8 - 1.2$ to simulate the influence of the dynamic milieu factors on the report quality.

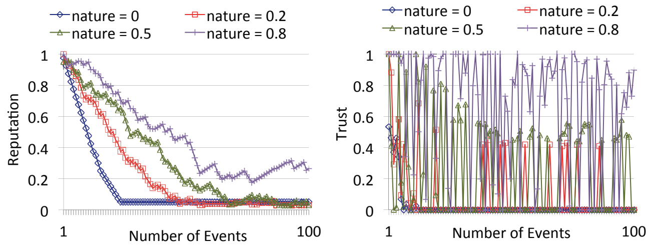
6.3 False Positive and False Negative Rates

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 reports received from a participant with different nature for various alarm thresholds and the results are shown in Table 4. 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 provenance, but definitely within an acceptable range.

Table 4 only shows the false positive and false negative rates under the default parameter settings. One

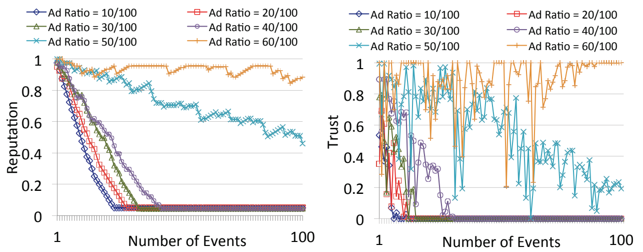
TABLE 4
False positive rates and false negative rates with default settings

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

Fig. 5. Impact of an adversary's nature on reputation and trust



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

Fig. 6. Impact of adversary ratio on reputation and trust

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.

6.4 Impact of Adversary's Nature

First, we want to see how an adversary's nature affects 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 5 (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 reputation feedback levels to be relatively small in order to prevent that one single task affects the reputation too much.

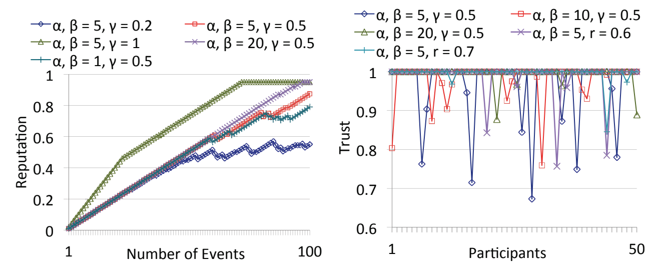
Next, we examine the computed trust values of the sensing reports sent by adversaries. Figure 5 (b) shows the result. The *0-nature* curve indicates that reports from an adversary with nature of 0 have a non-zero trust at the beginning when the reputation is still high, and the curve stays at 0 after 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, the higher nature an adversary has, the higher probability that his/her reports will get a high trust value.

6.5 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 by setting the number of adversaries as 10 to 60, out of 100 participants.

The result for the reputation updates is shown in Figure 6 (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 let the adversaries collude and their reports gain more supports from each other. When there is more adversaries than good participants, an adversary could maintain a high reputation level. However, as long as good participants are more than adversaries in the network, an adversary's report will get a negative reputation feedback with a high probability. Even when 40 out of 100 participants are colluding, their reputation still keeps decreasing until reaching a level close to 0.

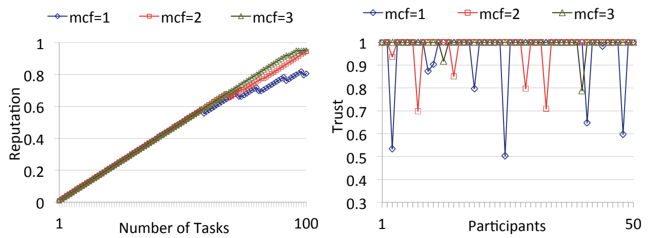
Again, for the same settings, we test the trust assessment and our result is shown in Figure 6 (b). The curves follow the similar trend as Figure 6 (a). However, the trust curves fluctuate much more than the reputation curves, which is the expected result. The reason is the contextual provenance and the similarity factor affect the trust of an individual report much more than they would affect the overall reputation. The randomness of these factors in our test makes the trust values of two consecutive sensing reports from the same adversary



(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. 7. Impact of α, β and γ on reputation and trust



(a) Reputation of a good participant with varying mcf

(b) Trust of sensing reports from 50 good participants with varying mcf

Fig. 8. Impact of mcf on reputation and trust

may differ a lot. This is particularly obvious when the ratio of colluding adversaries are high.

6.6 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. 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 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. We examine how different α, β and γ values would affect the reputation updates. As shown in Figure 7 (a), 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 provenance and therefore the good participant always gets positive feedback. When α, β becomes larger or γ becomes smaller, the randomness of the contextual provenance starts to appear. Hence, a portion of the sensing reports may get negative feedback due to the negative impacts from the contextual provenance. If the application is sensitive to the context, it is expected that reports with an unideal contextual provenance decrease the senders' reputation. That is why the reputation of a good participant goes up and down on some curves.

To show the impacts of α, β and γ on individual sensing reports clearly, we look at one task and we let 50 good participants that have a reputation of one at random sensing locations and times send their sensing reports. Figure 7 (b) shows how α, β and γ affect the trust of these sensing reports. It is clear large α and β magnify the impacts of the randomness of location and time factors. When γ is large, the similarity factor has bigger influence on the trust and this 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 carefully chosen and a proper γ value needs to be set in order to prevent the similarity factor from having too little or too much influence.

6.7 Impact of Maximum Cloaking Factor

We use a similar approach to test how the maximum cloaking factor (mcf) affects out trust and reputation assessment. When mcf is larger, participants may use larger cloaking area or cloaking interval for their reports to achieve a better location privacy. However, in this case, there is higher uncertainty involved in the location distance and time gap. Based on Eqn 1 and Eqn 2, the location distance factor or time gap factor becomes small when there is high uncertainty and thus we rely more on the other milieu factors and the similarity factor. This is why we observe less fluctuation on both reputation and trust evaluation caused by the location and time in Figure 8. Therefore we can conclude that better location privacy leads to less accuracy in the reputation and trust evaluation. mcf should be carefully chosen in order to maximize the capability of participants to cloak their location or time while get accurate trust and reputation assessments.

7 DISCUSSION

7.1 Anonymity Set

Our system enforces the application server to follow the protocol. Each sensing task is published to all participants. All participants are free to participate in any tasks and are expected to receive an RFC whenever they contribute. Therefore, we eliminates the possibility that the application server becomes malicious in terms of functionality and intentionally limits (or partitions) a task assignment to a single participant, thereby eliminating the k-anonymity. However, we do realize that our solution is depending upon a redundant number of participants. Like most of the other k-anonymity based privacy protection schemes, the size of the anonymity set is crucial and it works well only if the user base is large so that there is a redundant number of participants who have the same reputation levels. Due to such reason, our approach works the best for sensing tasks which require fairly large number of participants in a particular area, for example, traffic sensing.

For a system with a large user base, assuming majority of the users are good participants including both new or longtime users, there should be a redundant number of users with reputation levels from average to high. Therefore, we argue that the anonymity of good and new participants can be well protected by using our approach. Data has shown that a number of commercial mobile sensing applications like Gigwalk [1] and Waze [3] have already gained huge user bases and they are still undergoing a big growth [34]. We believe more similar applications with even larger user bases are soon going to be emerged.

7.2 Sybil Attacks

Many reputation systems are vulnerable to Sybil attacks, i.e., an attacker obtains multiple identities. The main incentive for a Sybil attack in traditional reputation systems like eBay is to have the multiple identities collectively promote each other's reputation. However, there is no direct interactions between users in mobile sensing applications. Therefore, Sybil attacks cannot take advantage of the mutual ratings.

Since users do not interact with each other in mobile 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.

To further mitigate Sybil attacks, the user registration process needs to enforce people to provide some scarce resources they process in order to get their unique credentials, so that people cannot freely register unlimited number of accounts. For mobile application scenarios like what we are considering in this paper, a good choice of unique resource that are required could be a phone number or an unique mobile device IMEI number.

8 Conclusion

Trust and anonymity are two conflicting objectives in a mobile sensing application. In this work, we proposed the ARTSense framework to achieve both of them at the same time without requiring a trusted third party. 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 of the usage of Blind IDs. 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 prototype implementation on Android shows that it only requires minimal computational overhead to run ARTSense on mobile devices. Our simulation results confirmed that with proper choices of the system parameters, different mobile sensing applications can be accommodated, and both user reputation and data trust can be accurately captured.

9 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] "Waze," http://www.waze.com/.

[4] B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan, and S. Madden, "CarTel: a distributed mobile sensor computing system," in *Proceedings of ACM SenSys*, 2006, pp. 125–138.

[5] R. Rana, C. Chou, S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: an end-to-end participatory urban noise mapping system," in *Proceedings of ACM/IEEE IPSN*, 2010, pp. 105–116.

[6] S. Eisenman, E. Miluzzo, N. Lane, R. Peterson, G. Ahn, and A. Campbell, "BikeNet: A mobile sensing system for cyclist experience mapping," *ACM Transactions on Sensor Networks*, 2009.

[7] L. Deng and L. Cox, "Livecompare: grocery bargain hunting through participatory sensing," in *Proceedings of ACM HotMobile*, 2009, pp. 1–6.

[8] 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.

[9] A. Kapadia, D. Kotz, and N. Triandopoulos, "Opportunistic sensing: Security challenges for the new paradigm," in *Proceedings of IEEE COMSNETS*, 2009.

[10] 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.

[11] 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.

[12] 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.

[13] K. Shilton, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Participatory privacy in urban sensing," in *Proceedings of the MODUS Workshop*, 2008.

[14] K. Huang, S. Kanhere, and W. Hu, "Preserving privacy in participatory sensing systems," *Computer Communications*, vol. 33, no. 11, pp. 1266–1280, 2010.

[15] 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.

[16] E. De Cristofaro and C. Soriente, “PEPSI: Privacy-enhanced participatory sensing infrastructure,” in *Proceeding of ACM WiSec*, 2011.

[17] A. Boukerch, L. Xu, and K. EL-Khatib, “Trust-based security for wireless ad hoc and sensor networks,” *Computer Communications*, no. 30, pp. 2413–2427, 2007.

[18] A. Dua, N. Bulusu, W. Feng, and W. Hu, “Towards trustworthy participatory sensing,” in *Proceedings of USENIX HotSec*, 2009.

[19] X. Wang, K. Govindan, and P. Mohapatra, “Collusion-resilient quality of information evaluation based on information provenance,” in *Proceeding of the IEEE SECON*, 2011.

[20] K. L. Huang, S. S. Kanhere, and W. Hu, “Are you contributing trustworthy data? the case for a reputation system in participatory sensing,” in *Proceedings of ACM MSWiM*. ACM, 2010, pp. 14–22.

[21] ——, “A privacy-preserving reputation system for participatory sensing,” in *Proceedings of IEEE LCN*. IEEE, 2012, pp. 10–18.

[22] D. Christin, C. Roßkopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, “Incognisense: An anonymity-preserving reputation framework for participatory sensing applications,” in *Proceedings of IEEE PerCom*, 2012.

[23] Q. Li and G. Cao, “Providing privacy-aware incentives for mobile sensing,” in *Proceedings of IEEE PerCom*, vol. 18, 2013, p. 22.

[24] X. Wang, W. Cheng, P. Mohapatra, and T. F. Abdelzaher, “ARTSense: anonymous reputation and trust in participatory sensing,” in *Proceeding of IEEE INFOCOM*, 2013.

[25] I. Krontiris and N. Maisonneuve, “Participatory sensing: The tension between social translucence and privacy,” *Trustworthy Internet*, 2011.

[26] X. Wang, K. Zeng, K. Govindan, and P. Mohapatra, “Chaining for securing data provenance in distributed information networks,” in *Proceeding of IEEE MILCOM*, 2012.

[27] D. Christin, A. Reinhardt, S. Kanhere, and M. Hollick, “A survey on privacy in mobile participatory sensing applications,” *Journal of Systems and Software*, 2011.

[28] C. Beecks, M. Uysal, and T. Seidl, “A comparative study of similarity measures for content-based multimedia retrieval,” in *Proceeding of IEEE ICME*, 2010.

[29] D. Chaum, “Blind signatures for untraceable payments,” in *Proceeding of CRYPTO*, 1982, pp. 199–203.

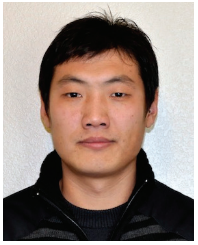
[30] R. Henry and I. Goldberg, “A Survey of Anonymous Blacklisting Systems,” *Technical Report, Centre for Applied Cryptographic Research, University of Waterloo*, 2010.

[31] P. P. Tsang, M. H. Au, A. Kapadia, and S. W. Smith, “Blacklistable anonymous credentials: Blocking misbehaving users without ttps,” in *Proceedings of ACM CCS*, 2007, pp. 72–81.

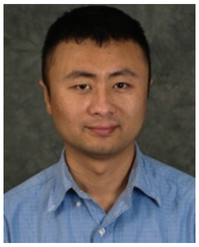
[32] “PowerTutor,” http://powertutor.org/.

[33] S. Goldwasser and M. Bellare, “Lecture notes on cryptography,” *Summer Course Lecture Notes at MIT*, 1996.

[34] “The Most Disruptive Companies In 2012,” http://blogs.sap.com/innovation/innovation/the-most-disruptive-companies-in-2012-024015.



Xinlei (Oscar) Wang received his B.E. degree in Electrical and Electronic Engineering at Nanyang Technological University, Singapore in 2008. He joined the Ph.D. program of the Department of Computer Science at University of California, Davis in 2009 and he is currently a Ph.D candidate. He is the recipient of the Best Graduate Researcher Award 2013 in the Computer Science Department. He has worked in the Science Outreach for Army Research (SOAR) program at U.S. Army Research Lab in 2012 and in the Energy and Sustainability Lab of Intel Labs in 2013. His research interests include information security and privacy in mobile and social networks, trust and reputation management in distributed systems, and data provenance and its impact on Quality of Information (QoI).



Wei Cheng received his Ph.D. degree in Computer Science from the George Washington University in 2010, and B.S. degree in Applied Mathematics, M.S. degree in Computer Science both from National University of Defense Technology, China, in 2002 and 2004. Currently, he is an Assistant Professor at Virginia Commonwealth University. He has worked as a Postdoc Scholar at University of California Davis. His research interests span the areas of Ubiquitous Computing and Cyber-Physical Networking System. In particular, he is interested in security, localization, cognitive technology, smartphone applications, Underwater Networks, and RFID Systems on Roads.

Prasant Mohapatra is a Professor in the Department of Computer Science at the University of California, Davis. He is currently serving as the Interim Vice-Provost and the CIO of UC Davis. He was the Department Chair of Computer Science during 2007-2013, and held the Tim Bucher Family Endowed Chair Professorship during that period. He received his doctoral degree from Penn State University in 1993, and received an Outstanding Engineering Alumni Award in 2008. In the past, he has been on the faculty at Iowa State University and Michigan State University. He has also held Visiting Scientist positions at Intel Corporation, Panasonic Technologies, Institute of Infocomm Research, Singapore, and National ICT Australia. He has been a Visiting Professor at the University of Padova, Italy and Yonsei University, South Korea. He was/is on the editorial board of the IEEE Transactions on Computers, IEEE Transactions on Mobile Computing, IEEE Transaction on Parallel and Distributed Systems, ACM WINET, and Ad Hoc Networks. He has been a Guest Editor for IEEE Network, IEEE Transactions on Mobile Computing, IEEE Communications, IEEE Wireless Communications, and the IEEE Computer. His research interests are in the areas of wireless networks, sensor networks, Internet protocols, and QoS. He is a Fellow of the IEEE and a Fellow of AAAS.

Tarek Abdelzaher is currently a Professor and Willett Faculty Scholar of the Department of Computer Science at University of Illinois at Urbana Champaign. He received his Ph.D. from the University of Michigan, Ann Arbor, in 1999. He was an Asistant Professor at the University of Virginia from August 1999 to August 2005. His interests lie primarily in systems, including operating systems, networking, sensor networks, distributed systems, and embedded real-time systems. He is Editor-in-Chief of the Journal of Real-Time Systems, an Associate Editor of the IEEE Transactions on Mobile Computing, the ACM Transaction on Sensor Networks, the International Journal of Embedded Systems and the Ad Hoc Networks Journal, as well as Editor of ACM SIGBED Review. He was Guest Editor for the Journal of Computer Communications and the Journal of Real-Time Systems, and is Co-Editor of IEEE Distributed Systems Online. He served on numerous technical program committees in real-time computing, networking, quality of service, distributed systems, sensor networks, multimedia, and mobile computing, among others. He also held organization positions for several conferences including RTAS, IPSN, RTSS ICDCS, DCoSS, and SenSys. Tarek Abdelzaher is a member of IEEE and ACM.