FIDES: A Trust-based Framework for Secure User Incentivization in Participatory Sensing

Francesco Restuccia and Sajal K. Das

Department of Computer Science
Missouri University of Science and Technology, USA
{frthf, sdas}@mst.edu

***Abstract*—Participatory sensing (PS) has recently attracted tremendous attention given its potential for a wide variety of sensing applications. Due to the fact that PS systems rely completely on the data provided by the users, incentivizing users' active participation while guaranteeing data reliability is paramount to effectively employ PS systems in practical scenarios. In this paper, we first define a set of attacks which compromise data reliability of existing PS applications. Next, we propose a scalable and secure trust-based framework, called FIDES, which relies on the concept of mobile security agents (MSAs) and Jøsang's trust model to rule out incorrect reports and reward reliable users. By simulating the FIDES framework on mobility traces of taxi cabs in San Francisco, we demonstrate that FIDES secures the PS system from the proposed attacks, guarantees high data reliability, and saves significant amount of revenue with respect to existing reward mechanisms.**

I. Introduction

Given its potential for a large variety of real-life applications, participatory sensing (PS) [1] has recently gained tremendous attention from the research community. PS is a sensing paradigm that allows ordinary citizens to participate in large-scale sensing surveys by using user-friendly applications installed in their smartphones. In this way, fine-grained sensing information is obtained from smartphone users without employing fixed and expensive infrastructure.

Figure 1 shows a typical architecture of a PS system. After operations such as data collection, filtering and aggregation, global information about the sensing area is sent back to the users through the PS application, so as to be used for their daily activities such as choosing the best route, or dress according to the current weather conditions. Real-life applications, which can take advantage of both the low-level sensor data and high-level user activities, range from real-time traffic monitoring applications like *Nericell* or *Waze* [2], [3] to air pollution or garbage monitoring [4]–[6] and social networking [7]. For an excellent survey of applications based on the PS paradigm, please refer to [8].

There are some unique characteristics that make PS substantially different from traditional sensing paradigms based on wireless sensor networks. By definition, PS is a *user-centric* paradigm, in the sense that PS applications rely completely on the users' willingness to participate and submit to the system accurate and up-to-date information. This implies that incentivizing users' participation and assessing the reliability of reported data becomes of fundamental importance in PS applications.

978-1-4799-4786-7/14/$31.00 ©2014 IEEE

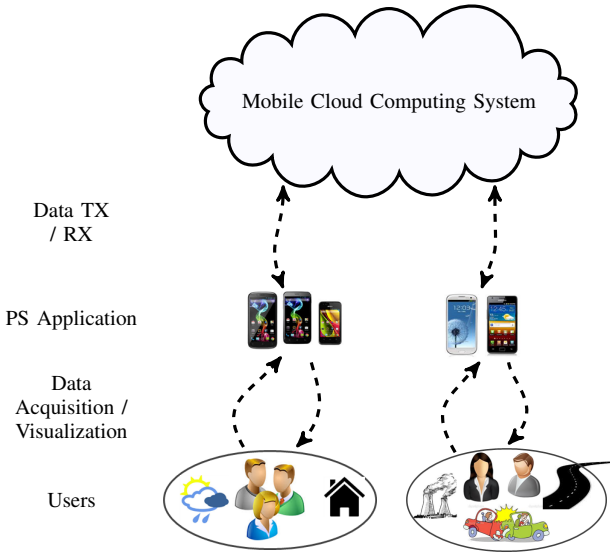

Fig. 1: Typical PS architecture.

To this end, in order to ensure *freshness* of data and avoid churning from the PS application, credit-based reward mechanisms (RMs) have been recently proposed in the literature [9]–[12]. The underlying idea is to reward users for their sensing services depending on their degree of participation. As far as the *reliability* aspect is concerned, trust-based frameworks [14], [15] based on user and contextual provenance have also been proposed to estimate the accuracy of reported data. In addition, location verification techniques based on ad-hoc networking [16], [17] have been proposed to validate the current location of users and therefore improve the overall reliability of the PS system.

Although existing work investigated how to incentivize user participation, keep track of user reputation, and verify the position of users, the following security flaws of PS systems still remain as open challenges.

- Current reward mechanisms [9]–[12] reward users *regardless* of the reliability (i.e., freshness and accuracy) of the data being sent. This implies malicious and/or unreliable users are rewarded for their sensing services irrespective of the accuracy and freshness of the data being sent. In addition, existing location verification methods [16], [17] are not able to estimate data reliability.
- Existing trust-based frameworks [14], [15] are not protected from attacks based on the spoofing of the current global positioning system (GPS) coordinates. In fact, by using widely available applications like *LocationHolic* or *FakeLocation* [18], it becomes extremely simple for attackers from any region of the world to fake out their location and gain reputation inside the system, thus exploiting the reward mechanism (if used) or feeding false data to the PS applications.

Assessing data reliability in an efficient way is extremely challenging in PS systems, mainly due to the highly distributed nature of the PS paradigm (up to hundreds of thousands of users [8]), and potentially large number of malicious users. To the best of our knowledge, no existing work has yet attempted to solve at once the problem of incentivizing users' participation *and* guaranteeing reliability of sensed data in PS systems taking into account GPS-spoofing-based attacks. This motivates our work.

In this paper, we first formalize a set of attacks which compromise existing reward systems and trust-based frameworks. Next, we propose FIDES[1], a novel trust-based framework that incentivizes users' active participation in PS applications and prevents the attacks modeled in this paper. FIDES uses a customized version of Jøsang's *opinion and trust* model [19] to deal with the uncertainty of user reputation, and relies on the concept of *mobile security agent* (redefined by us in this PS context) to help the PS applications assess the user's current reputation level in a scalable and secure manner.

To summarize, this paper makes the following novel contributions.

- We define a set of attacks to existing PS systems aimed at undermining existing reward systems, trust-based frameworks, and location-verification systems.
- We propose the FIDES framework to incentivize user participation, guarantee data reliability to the PS application, and solve the attacks defined in this paper. Additionally, we redefine the concept of *mobile security agent* (MSA) in PS applications to build user reputation.
- We extensively evaluate our proposed framework via simulation experiments on real-world mobility traces of taxi cabs in San Francisco [20], and calculate the amount of revenue saved by FIDES with respect to the RM presented in [11]. Results show that FIDES scales linearly with the number of users; it is efficient and incentivizes users' participation yet secures the PS application from attackers; allows the system administrator to save a large amount of revenue (thousands of revenue units with respect to [11]) and achieve high reliability levels of sensed data.

[1]FIDES is the latin word for *faith* and *trust*, and is the name of the Roman goddess of trust.

The rest of the paper is organized as follows. Section II introduces the system and threat models considered in this paper, while Section III summarizes the related work. Section IV describes in details the FIDES framework and Section V presents the simulation results. Section VI draws conclusions with directions of future work.

II. System and Threat Model

As shown in Figure 2, the PS system consists of a PS *platform* (PSP) hidden inside the cloud, and by PS *users*, which are directly connected to the PSP through 3/4G or WiFi Internet connection and run the PS application on their smartphones. In order to collect data, the PSP periodically broadcasts a request through the PS application to obtain users' sensed data. Then, users who choose to participate send their sensed data to the PSP through the Internet. Next, the information retrieved is processed by the PSP and eventually sent back to the users to be displayed on their smartphone screens. The application is responsible for handling data acquisition, transmission, and visualization, and is distributed to the users' mobile devices through common application markets (like *AppStore* or *Google Play*). Eventually, the PS system may use RMs such as [9]–[12] to incentivize user participation, and a trust-based system like [14], [15] to estimate the reliability of users. Note that this application scenario is very general, and can be adapted to many different types of PS applications.

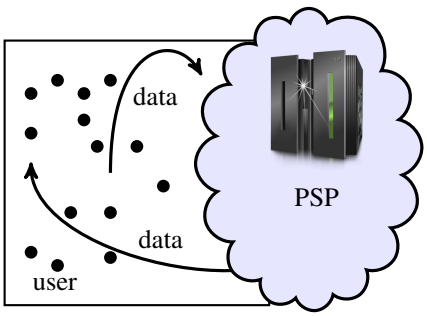

Fig. 2: System model.

A. Threat Model

As regards to the platform side, we consider the PSP trustworthy in terms of its functionality (such as user registration, issuing credentials, receiving, processing, and redistributing data). Furthermore, we suppose that confidentiality, integrity, and non-repudiation are addressed by using standard techniques such as cryptography and digital signatures. Like in Waze [3] and similar systems, we suppose users are identified by the PSP by username

TABLE I: Summary of Related Work.

Paper	Contribution	Description
Li'13 [9]	Reward Mechanism	Privacy-aware Incentive Mechanism
Luo'12 [10]	Reward Mechanism	Fairness & Social Welfare Mechanism
Yang'12 [11]	Reward Mechanism	User & Platform-centric Mechanism
Koutsopoulos'13 [12]	Reward Mechanism	Optimal Reverse Auction Mechanism
Huang'10 [13]	Data reliability	Consensus-based Trust Framework
Wang'11 [14]	Data reliability	Provenance-based Trust Framework
Wang'13 [15]	Data reliability	Anonymous Reputation Trust Framework
Talasila'10 [16]	Location verification	Bluetooth-based Location Verification Mechanism
Talasila'13 [17]	Location verification	Pictures and Bluetooth-based Mechanism
FIDES [**this paper**]	RM & Data reliability	Trust-based Framework for Data Reliability

and password, therefore authentication is guaranteed (which means no replica / sybil attacks are possible).

Let us now formalize three possible threats to the system described above. As far as attackers are concerned, we will assume malicious users are rational entities, interested in feeding false information to the system and earning credits from the RM as a secondary target. Given the trust-based system is handled by the PSP, we also assume attackers cannot obtain any knowledge about their current reputation level. We define[2] the following attacks, ordered by increasing strength. For the sake of generality, in each attack we suppose that users can eventually fake their position using applications such as *FakeLocation*, therefore appear in a different location than the physical one.

1) *No-knowledge (NK) attack.* This attack supposes the malicious user cannot obtain any knowledge about the sensing area. The attack is made up by the following steps.
   (i) The attacker registers into the system; **(ii)** At each request, the attacker feeds random data to the application.
2) *Partial-knowledge (PK) attack*. The attacker has partial knowledge of the sensing area, because the attacker is physically present inside the sensing area or can obtain the information in some way (e.g., the Internet). The target is to build a high reputation level to distribute false information about the sensing area through the PS application. To summarize,
   (i) The attacker logs into the system; **(ii)** At each request, the attacker feeds incorrect data with probability p and correct data with probability $1 - p$.
3) *Seesaw (S) attack*. This attack is a strong improvement of the PK attack. As above, the attacker has somehow partial knowledge of the sensing area. However, after building high reputation, here the attacker *periodically* sends incorrect data for a period T_i and then correct data for a period T_c, aiming at feeding false data into the PS system yet keeping her reputation level high. In details,
   (i) The attacker logs into the system; **(ii)** For time period T_p, the attacker feeds correct data to the application; **(iii)** The attacker feeds the system with incorrect data for a period T_i and then correct data for a period T_c; **(iv)** The attacker repeats step **(iii)**.

We would like to point out here that hereafter we will focus only on the attacks defined above. Other threats, for example DoS-based attacks, are out of the scope of this paper.

III. Related Work

In this section, we first survey the most relevant work on reward mechanisms, location validation techniques, and trust-based systems as applied to PS applications, as well as investigate the feasibility of the attacks defined in the last section.

As regards to RMs, Li and Cao [9] proposed a privacy-aware incentive scheme in which users earn credits by contributing data without leaking which data it has contributed and dishonest users cannot abuse the system to earn unlimited amount of credits. Luo and Tham [10] proposed two reward mechanisms that promote fairness and social welfare. Yang *et al.* [11] designed two incentive mechanisms using respectively Stackelberg game and auction-based incentive mechanism, and prove that both methods are truthful and profitable. Finally, in [12] Koutsopoulos proposed a mechanism for user participation level determination and payment allocation which minimizes the total cost of compensating participants, while delivering a certain quality of experience to service requesters.

Despite the soundness, the above RMs neglected to consider data and/or location reliability. Therefore, all the attacks described in the previous section are feasible in such RMs. In Section V, we will analyze in detail the damage in terms of revenue loss that attackers could cause to the PS administration by taking into consideration the RM presented in [11].

As far the validation of the users' location is concerned, in [16] Tilapia *et al.* proposed a scheme in which users are required by a provider to establish Bluetooth connections with nearby users at each sensing location to verify the actual location of users. A similar trust-based location validation system was proposed on [17]. In this paper, the trust in the system is bootstrapped by validating a small number of photos submitted by participants. Based on these validations, the location of these photos is assumed to be

[2]Although similar attack models may exist for other security applications, in the context of participatory sensing, such attacks are defined for the first time in this paper.

trusted. Then, the trust is extended to co-located sensed data points found in the Bluetooth range of the devices that provided the validated photos. Irrespective of the accuracy of the approaches, [16] and [17] did not consider the impact of the accuracy of the sensed data in assigning trust to users. Therefore, such solutions are prone to the attacks defined in the previous section.

There exists a plethora of literature dealing with reputation and trust in mobile ad-hoc networks or wireless sensor networks. For an excellent survey, refer to [21]. These works mainly aim at detecting or predicting any malicious behaviors and thus enhancing the overall security of the network [22]–[24]. Information-wise, techniques such as trust aggregation and/or data fusion are often used to quantify the reliability of the sensed data [25]. However, these approaches are not directly applicable to PS applications, in which rational, malicious behaviors like collusion and/or gain of reputation to exploit the system are concrete threats.

Only very recently the issue of establishing trust in PS applications has been considered. Specifically, in [13] the authors proposed a trust-based framework which computes the reputation of users using consensus-based techniques and Gompertz function. In [14], Wang *et al.* assess trustworthiness of information by calculating the similarity of information about the same event received from different users, the path difference and the current location of users. Also, they prove that their solution is resilient to some attacks based on collusion. In [15], the authors propose *ARTsense*, a reputation-based framework to handle data uncertainty in PS applications taking into account the aspect of anonymity as well. Similarly to [14], reputation levels in [15] are calculated using location and time-based assumptions.

Although the scheme yields very good accuracy, the consensus-based technique used in [13] to compute the reputation is prone to collusion-based attacks. In addition, given users' trustworthiness is estimated based on the location, the solutions in [14], [15] are prone to GPS-spoofing-based attacks, and therefore are not able to provide guarantees on data reliability. Conversely, FIDES uses an approach which is location-independent and immune to collusion-based attacks (as shown in the next section). Finally, such frameworks did not propose any reward mechanism, which is instead done in this paper.

To the best of our knowledge, FIDES is the first trust-based framework specifically targeted for PS that solves at once the problems of incentivizing users' active participation and guaranteeing data reliability in PS systems. In addition, we believe this is the first paper to use the notion of mobile security agents to secure PS applications.

IV. The FIDES Framework

In this section, we propose the FIDES framework to incentivize participation of honest users and prevent attacks against PS applications. After introducing some preliminary concepts behind the proposed framework, the concept of mobile security agent is illustrated. Finally, we detail how FIDES updates user reputation and handles user rewards.

A. Preliminaries

Definition 1. The reputation level r_i^j of the user u_i at a given time t_j is a probability measure of the reliability level of user u_i at time t_j. The value of r_i^j is 0 when u_i is completely unreliable and 1 when completely reliable. □

Definition 2. The quantity $p_i^j \in \mathbb{R}$ represents the amount of payoff paid to the user u_i at a given time t_j. □

Due to the incomplete and fuzzy knowledge about the reliability of each user and his/her data, we use Jøsang's trust model [19] to model and manipulate the uncertainty about user reputation. More precisely, in Jøsang's model, a belief metric, denoted by *opinion*, is used to express the degree of the belief in the truth of a statement. In the following, we will use opinions to formalize and update the user reputation over time. The definition of opinion is given as follows.

Definition 3. [19] An opinion ω is a vector $w = [b, d, u]$ satisfying $b + d + u = 1,\ \{b, d, u\} \in [0, 1]^3$ where the first, second and third component correspond to belief, disbelief and uncertainty respectively. □

In order to project onto a 1-dimensional probability space and produce a probability expectation value, we define the expected value $\mathbb{E}\{\omega\}$ of the opinion ω as

$$\mathbb{E}\{\omega\} = \frac{b + u}{b + d + 2u} \tag{1}$$

Finally, we will use the aggregation operator defined as follows.

Definition 4. [19] Let ω_p and ω_q be the opinions about two binary statements p and q. The *conjunction* $\omega_{pq} \triangleq \omega_p \oplus \omega_q$ of ω_p and ω_q, representing the opinion about both p and q being true, is defined as

$$\omega_{pq} = \begin{bmatrix} b_p \cdot b_q \\ d_p + d_q - d_p \cdot d_q \\ b_p \cdot u_q + b_q \cdot u_p + u_p \cdot u_q \end{bmatrix} \tag{2}$$

where $u_s = u_p + u_q + u_p \cdot u_q$. Conjunction of opinions is commutative and associative and requires independent arguments so that the conjunction of an opinion with itself is meaningless. When applied to opinions with zero uncertainty, it is the same as serial multiplication of probabilities. When applied to opinions with absolute belief or disbelief (i.e., $b = 1$ or $d = 1$), it produces the truth table of logical binary AND. □

Definition 5. Quantity x *matches* quantity y if $|x - y| \leq \theta_m$, where θ_m is a constant. □

Finally, we define as *sensing round* the set of operations performed by FIDES at each time step, where a *time step* is defined as the time interval between two subsequent sensing requests by the application (denoted by T_s).

B. Mobile security agents (MSA)

Let us now introduce the concept of a mobile security agent (MSA) redefined for PS applications. An MSA is defined as a user that periodically reports information about her surroundings to the PS platform, and considered to be *reliable* by the FIDES framework. By reliable we mean that FIDES assumes the information sent by MSAs is correct and reflects the real condition of the area nearby the MSA. In practical scenarios, the reliability of the MSAs may be guaranteed by some sort of reward provided upfront by the PS administration, which may be less advertisement in the PS application or additional credits by the RM. In Section V, we will evaluate the impact of different levels of reliability of the MSAs on the performance of FIDES. On the other hand, evaluating the impact of malicious MSAs performing collusion-based attacks is out of the scope of this paper and will be subject of future work.

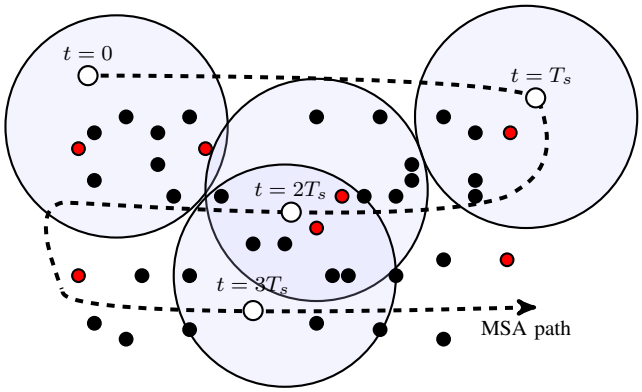

Fig. 3: An MSA roaming over the sensing area.

As for example, MSAs could be taxi cab drivers or bus drivers in urban sensing scenarios. In a campus activity monitoring application, on the other hand, MSAs could be students or police officers patrolling the campus.

Example. Figure 3 shows a sensing scenario in which only one MSA is roaming the sensing area. Legitimate and malicious users are depicted by black and red dots, respectively. Figure 3 also depicts the different locations over time of the MSA at different sensing rounds. At each time step, the MSA reports her sensed data to the PS platform (PSP). Then, such information is used by FIDES to update the reputation of the users nearby the MSA at that particular time. In particular, the reputation is updated only for the users inside a portion of sensing area, hereafter referred to as *update area* and depicted as a blue circle in Figure 3.

Let us now point out some advantages of using MSAs to evaluate the reputation of users.

- The size of the update area depends *only* on the particular type of PS application deployed[3]. Therefore, the number of users in the sensing area does *not* affect how the reputation of users is updated by FIDES. We will show in Section V.A that this property makes the FIDES framework *scalable* with the number of users of the PS application.
- FIDES assumes the MSAs as humans using a smartphone application. Therefore, given typical random mobility of MSAs, FIDES does *not* assume any particular mobility pattern, speed or trajectory of the MSAs.
- Conversely from consensus-based or location-based approaches, MSAs guarantee reliability through the evaluation of the accuracy of the *data* provided by the user. This gives FIDES resiliency to GPS-spoofing-based attacks. In fact, with this approach, the reliability of sensed data will be guaranteed *regardless* of the accuracy of users' position. In addition, the use of MSAs *prevents* collusion-based attacks by users, since FIDES relies only on reports coming from the MSAs to compute the reputation of users.

Also, it is worthwhile to point out that the optimal number of MSAs recruited by the PS administration will be strongly dependent on the particular PS application and the size of the sensing area. Future work will be devoted to find the optimal number of MSAs for a given sensing scenario.

C. Handling reputation and uncertainty

Let us now introduce the FIDES framework with the help of Figure 4. FIDES is designed as a middleware implemented between the users and the PS application, therefore we suppose that FIDES can access all user and application data. The first operation performed by FIDES is to initialize data structures for all users registered onto the system. Then, at each time step, the PS application sends a request to every user in the system to report their sensed data or impressions about their current surroundings. During this phase, the MSAs report information about their surroundings to the PS platform by using the same PS application as the one used by users[4].

This information, joint with the frequency of reports sent by the user, is used to update her current reputation level. Based on such reputation level, the reward for each user is calculated, while sensing reports are analyzed and accepted into the system based on the reliability level of the user that originated the report. Such information is then rendered homogeneous and re-distributed to the users of the application based on the user reputation level (as explained in the next subsection).

Let us now detail on how the reputation is calculated and updated for each user. FIDES relies on two different

[3]For example, in an air pollution monitoring application, such area may be as large as a neighborhood of a city, while in a traffic reporting application the area affected may be as large as a city block.

[4]Recall that the users authenticate themselves onto the system using secure credentials (e.g., username and password), therefore the attackers cannot steal the identity of MSAs.

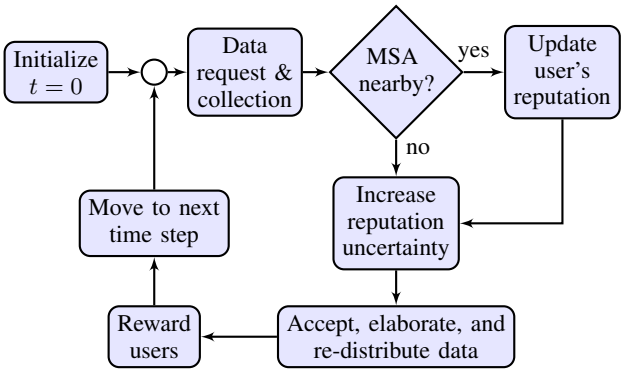

Fig. 4: Flow diagram of the FIDES framework.

opinions, respectively the *reliability opinion* $\omega^j_{r,i}$ and the *participation opinion* $\omega^j_{p,i}$, to characterize respectively the user reliability and user level of participation. Intuitively, such quantities keep track of reliability and level of participation at a specific time t_j. The reliability and participation opinions are initialized as follows for each user in the system, where r_0 and p_0 are constant values.

$$\omega^0_{r,i} = \begin{bmatrix} r_0 \\ 1 - r_0 \\ 0 \end{bmatrix} \omega^0_{r,i} = \begin{bmatrix} p_0 \\ 1 - p_0 \\ 0 \end{bmatrix} \tag{3}$$

Let us detail how such quantities are updated over time. As specified earlier, the PS application requests each registered user to send his/her report at each time step t_j. At a generic time step, the following operations are performed.

First, the sensed (and/or user opinion) data is collected from each participating user using the PS application. Second, the reputation of each user is updated based on the presence of the MSA in the surroundings of the users as follows. In case the MSA is nearby user u_i, the information sent by such user is cross-checked with the one sent by the MSA. Subsequently, each component of the reliability opinion is updated as follows.

- *Data match*. In this case, FIDES assumes the user has sent correct information about his/her surroundings to the application in the current sensing round. Then, the user reputation is updated as follows.

$$\omega^j_{r,i} = \begin{bmatrix} (b^{j-1}_{r,i})^{\alpha_r} & (d^{j-1}_{r,i})^{2-\alpha_r} & (u^{j-1}_{r,i})^{2-\alpha_r} \end{bmatrix}$$

  where α_r is a constant learning rate factor. In detail, the belief component is "intensified" by elevating it to the power of α_r, while the other two components are "de-intensified".
- *Data mismatch*. Conversely from previous case, FIDES assumes the user (or attacker) has sent incorrect information to the application in the current sensing round. Accordingly, the user reputation is updated by intensifying the disbelief component of the user reputation opinion and by de-intensifying the other two components.

$$\omega^j_{r,i} = \begin{bmatrix} (b^{j-1}_{r,i})^{2-\alpha_r} & (d^{j-1}_{r,i})^{\alpha_r} & (u^{j-1}_{r,i})^{2-\alpha_r} \end{bmatrix}$$

After the update, the three quantities are all normalized to sum up to one.

If no MSA is available, there is no information available to update the reliability level of the user. Therefore, FIDES intensifies the uncertainty on the reliability of the user and de-intensifies the other two components as follows.

$$\omega^j_{r,i} = \begin{bmatrix} (b^{j-1}_{r,i})^{2-\alpha_u} & (d^{j-1}_{r,i})^{2-\alpha_u} & (u^{j-1}_{r,i})^{\alpha_u} \end{bmatrix} \tag{4}$$

Regarding the participation opinion, FIDES keeps track of the number of times N^j_i user u_i has submitted her data to the application. Such quantity is divided by the number of time slots $T_j = \frac{t_j}{T}$ elapsed since $t = 0$ to keep track of the frequency of data submission over time. Specifically, the participating opinion $\omega^j_{p,i}$ is updated as

$$\omega^j_{p,i} = \begin{bmatrix} b^j_{p,i} = N^j_i / T_j \\ d^j_{p,i} = 0 \\ u^j_{p,i} = 1 - N^j_i / T_j \end{bmatrix} \tag{5}$$

After updating the reliability and participation opinions for each user in the system, the two opinions are combined to form the final opinion $\omega^j_{f,i}$ using the aggregation operation, i.e.,

$$\omega^j_{f,i} = \omega^j_{r,i} \oplus \omega^j_{p,i}$$

Finally, the reputation of the user is calculated using the expected value of the final opinion as described in Equation (1).

$$r^j_i = \mathbb{E}\{\omega^j_{f,i}\} \tag{6}$$

D. Trust assessment and reward mechanism

As anticipated, FIDES relies on the reputation level of users to exclude false reports from the PS application and at the same time, reward legitimate users for their sensing services. In particular, at each time step t_j, FIDES accepts into the system only reports originated from users having a reputation level greater than a pre-defined threshold, θ_r. That is, the report k_i originated from users u_i at time step t_j is accepted if and only if $r^j_i > \theta_r$. After assessing the trust level of each sensing report, up-to-date information about the sensing area is sent back to each user having reputation level greater than θ_r.

The final operation performed by FIDES is to reward users based on their reputation level. In particular, let N and U be the total number of registered users and the number of users having reputation level greater than θ_p, respectively. Let R be the (fixed) total amount of reward the PS application announces at every time step t_j to incentivize users. FIDES defines two payment strategies to reward users.

- *Fixed reward.* In this strategy, θ_r is set to θ_p, and each user having reputation level above threshold θ_p receives a payoff of $\frac{R}{N}$ credits (zero otherwise). Users

having reputation level below θ_r receive no credits. In details,

$$p_i^j = \frac{R}{N} \text{ if } r_i^j > \theta_p \quad (7)$$

- *Variable reward.* Conversely from the previous strategy, in the variable reward strategy users are paid proportionally to their reputation level as follows.

$$p_i^j = \frac{r_i^j}{\sum_{k=1}^{U} r_i^k} \cdot \frac{R \cdot U}{N} \text{ if } r_i^j > \theta_p \quad (8)$$

Users having reputation level below threshold θ_p receive no credit. As for the threshold θ_r, the choice between a fixed and a variable reward strategy depends on the particular context in which the PS application is distributed. In particular, the fixed reward allows more security, given $\theta_r = \theta_p$ and therefore more reports are ruled out. However, the variable reward strategy is more flexible and may better incentivize users, since users will be stimulated by the prospective of earning more credits by having a greater reputation value. Note also that the variable reward strategy may in general prevent churning from the system, since in this case more users will be paid (remember that $\theta_p < \theta_r$).

We want to point out that, at each time step, the total payoff paid by the application to the users *cannot* exceed the maximum value of $\frac{R \cdot U}{N}$. This is a *strict bound* on the reward given at each time step. In addition, note that the reward mechanisms described above are independent from the trust assessment performed by FIDES, and can be eventually modified and adapted to the actual sensing scenario, number of users, and/or budget of the PS administration.

V. Experimental Results

In this section, we evaluate through simulation experiments the performance of FIDES in terms of security and scalability. In order to simulate a realistic PS application scenario, we based our MSA mobility traces on the dataset of the *Cabspotting* project [20]. In this project, taxi cabs in San Francisco traveling through all the Bay Area were tracked for about two years, aiming at gathering data about their mobility. In order to keep track of the cabs' position over time, the vehicles were equipped with GPS devices, which would regularly send information about their position to a central server along with a timestamp. Therefore, it was possible for us to reconstruct each taxi's trajectory over space and time and use such trajectory to model the mobility of MSAs. Specifically, the sensing location is a 4×4 square kilometers area centered at GPS coordinates (37.7656, -122.4293) in Downtown San Francisco. For the sake of simplicity, we modeled the update area of MSAs as circles having the MSA as center and radius equal to 200m (to simulate a traffic monitoring application like Waze [3]).

Regarding the user mobility model, we did not use mobility traces but instead the Truncated Lévi Walk (TLW) mobility model [26], which has been shown to best represent the mobility of humans [27]. The TLW model assumes that humans have a constant speed depending on the mobility category they belong to (pedestrians, vehicles, bicycles, etc.). Due to space limitations, we refer the reader to [27] for additional insights.

A. Impact of system parameters and user participation

Let us demonstrate the impact of system parameters and user participation on the user reputation. As default system parameters, we chose as update parameters $\alpha_r = 0.3$ and $\alpha_u = 0.9$, $r_0 = p_0 = 0.5$, $\theta_r = 0.8$ and $\theta_p = 0.7$, while the number of users and MSAs equal to 1000 and 100, respectively. The confidence intervals are set to 95%. For the sake of graphical clarity, the confidence intervals are not shown when less than 1% of the average.

To evaluate the impact of the system parameters on users' reputation, Figure 5 shows the expectation of the reputation opinion of a user transmitting at each sensing round reliable information, as a function of α_r and α_u. Please recall from Section IV.D that, in case the user transmits reliable information only, the reputation decreases only by effect of the uncertainty. From Figure 5 we observe that a greater α_r corresponds to a lower user reputation level over time, whereas α_u impacts on how much the reputation decreases when the uncertainty increases. In fact, the reputation is much more variable in the case when $\alpha_u = 0.6$, implying that a lower α_u corresponds to a higher responsiveness of FIDES, but at the same time greater fluctuations of the reputation opinion. Therefore, α_u may be tuned by the PS administration based on the desired security level. In order to evaluate the impact of user participation on her reputation, in Figure 6 we depict the user reputation over time as a function of the probability p_t of transmission at each time step. Figure 6 remarks that FIDES is able to map well the participation level of users into different reputation level (thereby resulting in a greater reward to the most participating users), and incentivize users to submit regularly their data (so as to receive a greater reward).

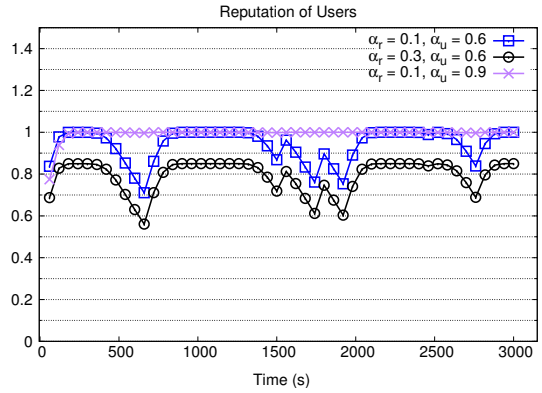

Fig. 5: User reputation (α_r and α_u).

As far as the scalability and efficiency issues are concerned, we now introduce a metric called *convergency time* to quantify the time necessary to good users to achieve high reputation levels. In detail, FIDES has a convergency

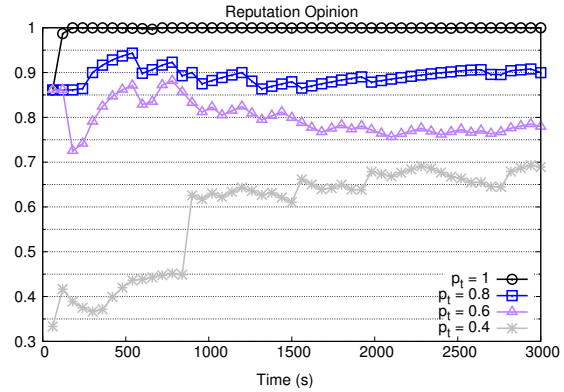

Fig. 6: User reputation (p_t).

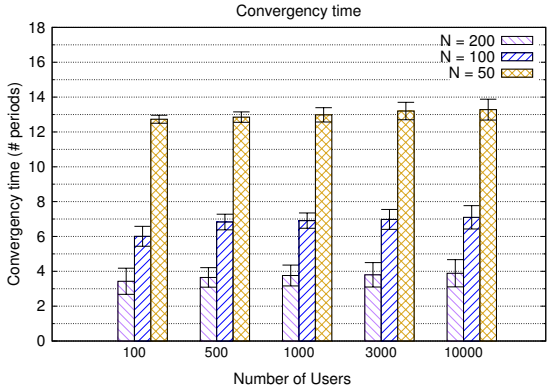

Fig. 7: Convergency time.

time of t_c if at time $t = t_c$ at least 90% of the good users have a reputation level at least θ_r. This metric is important because FIDES does not allow users to be paid unless they have a very high reputation level (see Section IV.D). Therefore, a lower convergence time indicates a lower risk of massive churning from the PS application by the PS administration[5]. Future work will be devoted to investigate the trade-off between the churning phenomenon and cost minimization by the PS administration (e.g., number of MSAs and amount of reward per user).

Figure 7 depicts the convergence time of FIDES (expressed in number of time steps) as a function of the number of MSAs deployed and the number of users in the sensing area. Note that Figure 7 also evaluates the impacts of the number of MSAs on the system. As expected, the convergence time decreases almost linearly as the number of MSAs increases. This is because the reputation level of users is updated faster when more MSAs are available.

However, the most important result concluded by Figure 7 is that the FIDES framework exhibits almost *constant* convergency time as the number of users increases. As mentioned in Section IV.B, this is due to the fact that the number of users present in the sensing area does not impact on how the reputation of users is calculated. This implies that FIDES is scalable and deployable in many real-world sensing scenario in which the number of users may assume significant values.

[5]This is because users may eventually get bored and leave the application if they do not get enough reward after some time.

B. Resilience from attacks

Let us now evaluate the impact of the attacks described in Section II with Figure 8, which shows the average reputation of all the attackers while executing different attacks. In particular, in the NK attack, all attackers start to send random data from $t = 0$. In the PK attack, attackers send reliable (resp. random) data with probability 0.7 (resp. 0.3) from $t = 0$. Finally, in the S attack, malicious users send reliable data until t = 1300s, then send unreliable data for the next 5 time steps, then resume sending reliable data. In these experiments, we considered a percentage of attackers of 20%.

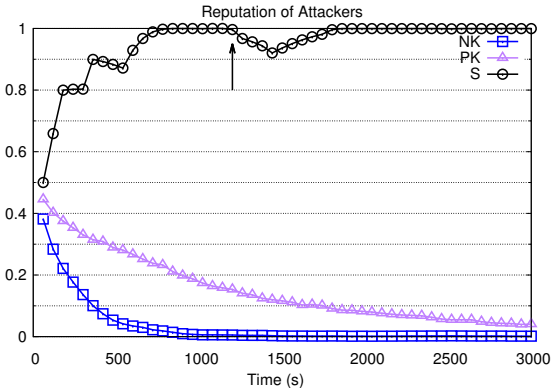

Fig. 8: Attackers reputation (p_a).

As regards the NK and PK attacks, from Figure 8 we conclude that FIDES is able to exclude unreliable reports very well by giving the attackers zero reputation after a reasonably short period of time. Regarding the S attack, we conclude the attackers' reputation still remains very high after 5 reports were incorrect. However, by appropriately setting $\theta_r = 0.985$, FIDES protects the PS system from the S attack as well. We conclude that FIDES is able to protect the PS system from all the attacks defined in Section II without compromising the functionality of the PS application.

It is worthwhile to remark here that FIDES accepts into the system only reports originated from users having reputation level greater than θ_r and pays off users above θ_p. Therefore, Figure 8 remarks that unreliable reports originated from attackers will be ruled out and attackers will *not* be rewarded by FIDES when they submit unreliable reports. Therefore, Figure 8 also concludes the reward paid to users submitting unreliable reports is **zero**, regardless of the type of attack being employed.

In order to evaluate the accuracy of FIDES as the percentage of attackers increases, we define the True Positive Ratio (TPR) as the ratio of the number of malicious users

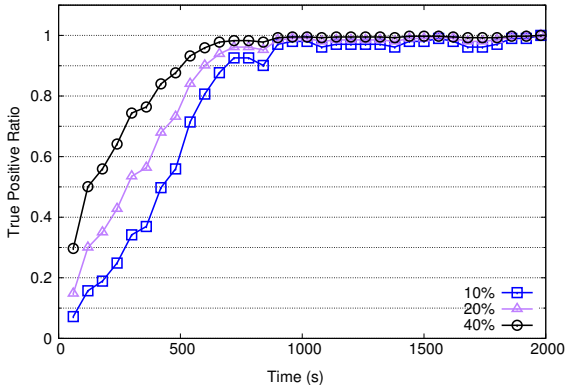

Fig. 9: True Positive Ratio.

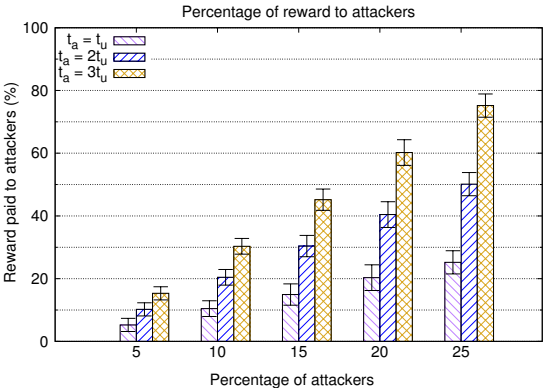

Fig. 11: Reward paid to attackers in [11].

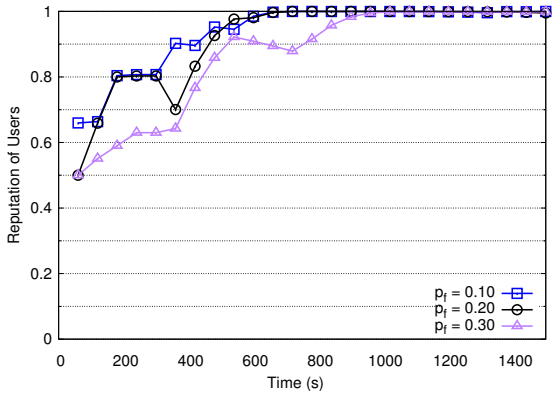

Fig. 10: User reputation (p_f).

detected by FIDES to the actual number of malicious users registered to the PS application. Figure 9 shows the TPR of FIDES as a function of time and percentage of attackers, supposing attackers submit false data to the system at each time step. The most important result that Figure 9 concludes is that the accuracy of FIDES is *independent* from the number of attackers in the system. Intuitively, this is because the MSAs update the reputation of each user independently. Note that this implies resiliency against collusion-based attacks as stated in the previous section.

Surprisingly, from Figure 9 it appears that FIDES was able to recognize attackers faster as the number of attackers increases. This can be explained as follows. At the beginning, every user in the system has low reputation level. As time passes, the reputation of malicious users decreases (due to fact that they are transmitting unreliable information), while the reputation of good users increases. Therefore, if the percentage of attackers is greater, the number of good users mistaken for malicious users will be less, making the TPR higher in this case. However, regardless of the number of attackers, after a reasonably low period of time, FIDES obtains TPR accuracy of nearly 100%, meaning that FIDES maintains high accuracy *regardless* of the number of attackers in the system. Furthermore, to evaluate how unreliable MSAs impact on the reputation of reliable users, Figure 10 shows the reputation of a user submitting correct data at each time step, in function of the probability p_f of the MSA to send incorrect data at each time step. As we can see, the user still reaches maximum reputation level after a reasonable amount of time regardless of the considered error rate of the MSA. This is because the effect of unreliable MSA reports is canceled out over time by reliable reports. This allows FIDES to be resilient to eventual hardware errors of the MSAs and demonstrates that FIDES can be implemented in practical scenarios where MSAs could be partially unreliable.

Finally, in order to understand how much revenue the administration of the PS application can be saved by using FIDES, we implemented and simulated the RM scheme due to Yang *et. al* [11]. In particular, we focused our attention on the Platform-Centric reward model, in which the users are paid proportionally to the time t_u they declare to dedicate to the sensing services (see [11] for additional details). We focused on this particular model because of its simplicity and its strong game-theoretical properties. In particular, we assumed t_u as a uniform random variable (r.v.) between 0 and 10s, while the attackers choose from an uniform r.v. t_a multiple of t_u (to obtain eventually more reward) and implement the NK attack. Figure 11 shows the average percentage of revenue that attackers receive from the RM in [11] at each time step with respect to the total number of reward R, as a function of the number of attackers and t_a. As expected, Figure 11 concludes that the amount of revenue the attackers steal from the RM grows linearly with the number of attackers. Remember from Figure 8 that FIDES does **not** reward attackers under any considered attack.

In order to have an idea of the impact of attackers in term of revenue loss in real-life PS application using [11] (and similar RMs), let us suppose to have a percentage of attackers equal to 5%, 1000 users in the system, $R = 10\$$, t_a equal to t_u, and a time step of $60s$. Every minute (i.e., time step), the attackers steal $0.5\$$ from the system, which means in a day the attackers steal $720\$$. In a month and a year, respectively, the administration will lose $21{,}600\$$ and

262,800$, respectively. If the time step was 10 minutes, the attackers would still steal 26,280$ a year. By distinguishing malicious from good users, FIDES prevents such losses and allows the PS administration to use this revenue to recruit more mobile security agents and eventually render the application even more secure.

VI. Conclusions

In this paper we have presented FIDES, a novel scalable, trust-based framework for incentivizing users and securing participatory sensing (PS) applications. First, we have defined three novel threats in existing reward mechanisms (RMs) and trust-based framework. Next, we have presented FIDES, which is a framework based on *Jøsang's opinion model* and the concept of *mobile security agent* (MSA) that incentivizes users yet secures the PS application from the attacks defined in this paper. Finally, we have extensively evaluated the scalability and security level provided by FIDES by using real-world mobility traces of taxi cabs in San Francisco. Results conclude that FIDES is a highly scalable and secure framework that secure existing PS system from the GPS-based attacks defined by us in this paper and therefore helps the PS administrations saving substantial revenue as compared to an existing RM.

As part of future work, we plan to find the minimum number of MSAs to recruit for a given sensing scenario through analysis. We also plan to develop a game-theoretic model of the FIDES framework, as well as implement a PS application like *Waze* [3] to test FIDES in real-life sensing scenarios. This will allow us to study the best reward strategy in different participatory sensing scenarios, thus finding the optimal solution that maximizes the PS administration's reward yet minimizes the risk of churning from the PS application.

References

[1] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, and M. B. Srivastava, "Participatory sensing", in Proc. of the 4th ACM Conference on Embedded Network Sensor Systems (SenSys), pp. 117-134, 2006

[2] P. Mohan, V. Padmanabhan, and R. Ramjee, "Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones", in Proc. of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys), pp. 323 - 336, 2008.

[3] Waze project. http://www.waze.com.

[4] M. Mun, S. Reddy, K. Shilton, N. Yau, J. Burke, D. Estrin, M. Hansen, E. Howard, R. West, and P. Boda, "Peir, the Personal Environmental Impact Report, as a Platform for Participatory Sensing Systems Research", in Proc. of the 7th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys), pp. 55-68, 2009.

[5] Intel/UC Berkeley, "Urban Atmospheres", http://www.urban-atmospheres.net/.

[6] CENS/UCLA,"Participatory Sensing / Urban Sensing Projects – http://research.cens.ucla.edu/.

[7] E. Miluzzo, N. D. Lane, K. Fodor, R. Peterson, H. Lu, M. Musolesi, S. B. Eisenman, X. Zheng, and A. T. Campbell, "Sensing meets Mobile Social Networks: The Design, Implementation, and Evaluation of the CenceMe Application", in Proc. of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys), pp. 337-350, 2008.

[8] N.D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A.T. Campbell, "A Survey of Mobile Phone Sensing", *IEEE Communications Magazine*, Vol. 48, No. 9, pp. 140-150, 2010.

[9] Q. Li, and G. Cao, "Providing Privacy-Aware Incentives for Mobile Sensing", in Proc. of the 11th IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 76-84, 2013.

[10] T. Luo, and C. Tham, "Fairness and Social Welfare in Incentivizing Participatory Sensing", in Proc. of the 9th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), pp. 425-433, 2012.

[11] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to Smartphones: Incentive Mechanism Design for Mobile Phone Sensing", in Proc. of the 18th ACM International Conference on Mobile Computing and Networking (MobiCom), pp. 173-184, 2012.

[12] I. Koutsopoulos, "Optimal Incentive-driven Design of Participatory Sensing Systems", in Proc. of the 32nd IEEE International Conference on Computer Communications (INFOCOM), pp. 1402-1410, 2013.

[13] K.L. Huang, S.S. Kanhere, and W.Hu, "Are You Contributing Trustworthy Data?: The Case for a Reputation System in Participatory Sensing", in Proc. of the 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM), pp. 1422, 2010.

[14] X. Wang, K. Govindan, and P. Mohapatra, "Collusion-resilient Quality of Information Evaluation Based on Information Provenance", in Proc. of the 8th IEEE Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON), pp. 395-403, 2011.

[15] X. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "ARTSense: Anonymous Reputation and Trust in Participatory Sensing", in Proc. of the 32nd Annual IEEE International Conference on Computer Communications (INFOCOM), pp. 2517-2525, 2013.

[16] M. Talasila, R. Curtmola, and C. Borcea, "Link: Location verification through immediate neighbors knowledge", in Proc. of the 7th International ICST Conference on Mobile and Ubiquitous Systems, (MobiQuitous), pp. 210-223, 2010.

[17] M. Talasila, R. Curtmola, and C. Borcea, "ILR: Improving Location Reliability in Mobile Crowd Sensing", International Journal of Business Data Communications and Networking, Vol. 9, No. 4, pp. 65-85, October-December 2013.

[18] LocationHolic and MyFakeLocation (http://www.locationholic.com and https://play.google.com/store/apps/).

[19] A. Jøsang, "An Algebra for Assessing Trust in Certification Chains", in Proc. of the Network and Distributed Systems Security Symposium (NDSS), 1999.

[20] M. Piorkowski *et al.*, "CRAWDAD Data Set Epfl/Mobility (v. 2009-02-24)", http://crawdad.cs.dartmouth.edu/epfl/mobility, Feb. 2009.

[21] H. Yu, Z. Shen, C. Miao, C. Leung, and D. Niyato, "A Survey of Trust and Reputation Management Systems in Wireless Communications", Proceedings of the IEEE, Vol. 98, No. 10, pp. 1755-1772, 2010.

[22] A. Boukerch, L. Xu, and K. EL-Khatib, "Trust-based Security for Wireless Ad Hoc and Sensor Networks", Computer Communications, Vol. 30, No. 18, pp. 2413-2427, 2007.

[23] M. J. Probst and S. K. Kasera, "Statistical Trust Establishment in Wireless Sensor Networks", in Proc. of the 13th International Conference on Parallel and Distributed Systems, pp. 1-8, 2007.

[24] C. Papageorgiou, K. Birkos, T. Dagiuklas, and S. Kotsopoulos, "Dynamic Trust Establishment in Emergency Ad Hoc Networks", in Proc. of the 2009 International Conference On Communications And Mobile Computing (IWCMC), pp. 26-30, 2009.

[25] Y. Sun, H. Luo, and S. K. Das, "A Trust-based Framework for Fault-tolerant Data Aggregation in Wireless Multimedia Sensor Networks", *IEEE Transactions on Dependable and Secure Computing*, Vol. 9, No. 6, pp. 785-797, Nov-Dec 2012.

[26] S. Hachem, A. Pathak, and V. Issarny, "Probabilistic Registration for Large-Scale Mobile Participatory Sensing", in Proc. of the 11th IEEE International Conference on Pervasive Computing and Communications (PerCom), pp. 132-140, 2013.

[27] I. Rhee, M. Shin, S. Hong, K. Lee, S. Kim, and S. Chong, "On the Lévy-Walk Nature of Human Mobility",*IEEE/ACM Transactions on Networking*, Vol. 19, No. 3, pp. 630-643, 2011.