The Lord of the Sense: A Privacy Preserving Reputation System for Participatory Sensing Applications

Antonis Michalas
Security Lab
Swedish Institute of Computer Science
Stockholm, Sweden
antonis@sics.se

Nikos Komninos
Department of Computer Science
City University London
London, United Kingdom
nikos.komninos.1@city.ac.uk

***Abstract*—Electronic devices we use on a daily basis collect sensitive information without preserving user's privacy. In this paper, we propose the lord of the sense ($LotS$), a privacy preserving reputation system for participatory sensing applications. Our system maintains the privacy and anonymity of information with the use of cryptographic techniques and combines voting approaches to support users' reputation. Furthermore, $LotS$ maintains accountability by tracing back a misbehaving user while maintaining k-anonymity. A detailed security analysis is presented with the current advantages and disadvantages of our system.**

***Index Terms*—Participatory Sensing, Distributed Sensing, Urban Sensing, Privacy, Anonymity, Security, Reputation Systems**

I. Introduction

In the past few years we have seen a dramatic growth in the use of mobile Internet, including the massive demands for mobile data, the growth of mobile video, and the rise of the smartphones and tablets as a new gateway to the web itself.

According to Cisco [1], in 2011 mobile data traffic was 8 times the size of the entire global Internet in 2000 (597 petabytes vs. 75 petabytes). It is also estimated that the number of mobile-connected devices will exceed the number of people on earth by the end of 2012. By 2016, there will be 1.4 mobile devices per capita and there will be over 10 billion mobile-connected devices, including machine-to-machine (M2M) modules. Again, the number will exceed the world's population at that time (7.3 billion).

The adoption of mobile devices in combination with the evolution of Web 2.0 has emerged to a new research era that is often referred to as *participatory sensing*. Participatory sensing targets the pervasive collection of information from the actual environment of the user [2].

Nevertheless, the entanglement of people in the process rises many security concerns. Furthermore, protection of user's privacy is an increasingly relevant topic since mobile devices gather and distribute sensored data from the user's actual environment without always asking for a permission. More precisely, technology copies and collects private data without users approval, such as tracking their location or harvesting their address book. Recent studies have shown that privacy issues discourages the user's involvement in participatory sensing and, thus, it has a great impact on the acceptability and adoption of these new technologies [2].

A. Our Contribution

This paper presents Lord of the Sense ($LotS$), a privacy-preserving reputation system for participatory sensing applications. In particular, users in $LotS$ manage to keep their real identity hidden, while at the same time they have a reputation score for which they cannot lie about or shed. The reputation of each user is updated and demonstrated in a way that does not compromise user anonymity. By using appropriate cryptographic mechanisms, unlinkability between the real identity of a user and the reports that she publishes is achieved. In addition, each user can vote for every published report while at the same time double voting is prevented.

B. Organization

The remainder of this paper is organized as follows. In Section II, we present the most important studies that deal with privacy issues in the field of participatory sensing, as well as protocols that will help us design a mechanism to protect the privacy of users in such applications. In Section III, we describe our problem statement and the basic terms we use in the rest of the paper. In Section IV, we present our main protocol, while in Section V we provide a security discussion in which we show the resistance of our protocol against numerous attacks. Finally, in Section VI we conclude the paper.

II. Related Work

Although there are many participatory sensing applications [3]–[13], only some of them deal with the problem of anonymous and private data reporting. Furthermore, the absence of reputation schemes that would create a rank for each user, based on their reports and the votes that each report receives from the community is even bigger.

AnonySense [14] allows a variety of applications to request sensor data using a flexible tasking language and later to receive by the system the sensor data from personal mobile devices. Data is collected in an opportunistic and delay-tolerant manner, in which a large and dynamic set of mobile nodes can volunteer to accept tasks and send back reports, both reliably and anonymously. Anonysense is based on a Mix Network [15] in order to guarantee user unlinkability between reports with respect to WiFi access points. While there exists extensive research on privacy, anonymity and unlinkability in WiFi networks,

such an assumption imposes severe limitations on the scope of participatory sensing applications, as an ubiquitous presence of openWiFi networks is neither realistic today nor anticipated in the near future. Therefore, such an assumption would heavily limit the applications' availability and accuracy.

E. De Cristofaro and C. Soriente introduced PEPSI [16], a privacy-enhanced participatory sensing protocol with main aim to hide reports and queries to unintended parties. Their protocol is based on Identity Based Encryption (IBE) [17] which enables non interactivity, a major concern in participatory sensing scenarios, where mobile nodes and queriers have no direct communication or mutual knowledge. Even though PEPSI is considered as a lightweight protocol, which makes it suitable for devices with limited resources, it has two main disadvantages. First, the security of the protocol relies on the assumption that the service provider is not colluding with either the registration authority or the queriers. Second, PEPSI needs to trust the network operator to remove sensitive data from reports before forwarding them to the service provider.

In [18] authors presents a location based privacy system ($LOCATE$) for participatory sensing applications that targets Android devices. $LOCATE$ allows users to locally sense and store data as well as issue queries on data stored across the system. The main differences between $LOCATE$ and its predecessors is mainly found in the fact that $LOCALE's$ user data is generated and stored locally on the individual user's device while in the existing approaches, data sharing operate under the assumption that user data is maintained in a centralized database.

To the best of our knowledge, IncogniSense [19] is the only privacy preserving reputation system designed for participatory sensing applications. It is an anonymity-preserving reputation framework based on blind signatures, which is agnostic to both the reputation algorithm and applications. IncogniSense preserves the anonymity of the users by using pseudonym systems. Nevertheless, IncogniSense has two main drawbacks. The first drawback is the absence of a voting procedure; a feature that makes the problem of building a privacy-preserving reputation system for participatory sensing applications more difficult and the protocol more concrete. The second drawback is related to the fact that IncogniSense does not provide any mechanism for accountability. More precise, when a user acts maliciously there is no way that her real identity will be revealed in order to encounter the repercussions that are defined from the community regarding misbehaving users.

III. Problem Statement & Definitions

Our protocol consists of clients, a registration authority (RA) and an application server which is referred as the community (C). Clients collect data from their mobile devices and submit them anonymously to the application server.

We assume that the reader is familiar with the concept of public key cryptography. For the needs of our protocol, each authority has generated a public/private key pair. The private key is kept secret, while the public key is shared with the rest of the community. These keys will be used to secure message exchanges in the community.It is also assumed that users knows the public keys of RA and C. Furthermore, our protocol also relies on the use of group signatures [20] for the verification of a user without revealing her real identity.

Registration Authority (RA): RA is responsible for the registration of users. Additionally, RA has a public/private key pair denoted as pk_{RA}/sk_{RA}. Apart from that, RA is responsible for generating parameters that will be used for the proper function of our protocol (submit a new report, vote for a report, update reputation scores).

Community (C): C is responsible for handling the main tasks of our protocol. More precisely, C allows a user to publish a report, to vote for an already published report and to update the reputation score of registered users. Additionally, C has a public/private key pair denoted as pk_C/sk_C.

Group Signatures: Groups signatures as described in [20] is a group of users that can create a signature group with the following properties:

- Only members of that group can sign message on behalf of the group
- The receiver can verify that signed message is a valid signature from that group
- The resulting signature does not reveal signer's identity
- If a user acts maliciously, the signature can be opened and reveal the real identity of the user

In this paper, we focus on the following problem:

Problem Statement: *Let $U = \{u_1, \ldots, u_n\}$ be the set of all users that are registered through a registration authority RA and C the community that handles all the reports. In addition to that, a reputation score r_{u_i} must be associated with each user. Lets suppose that a user u_i wishes to publish a report $R_j^{u_i}$ to the community. First, u_i must convince C that she is a registered user without revealing her real identity. Then, if u_i has been successfully verified, $R_j^{u_i}$ is published to C so that each user can see not only the information that $R_j^{u_i}$ contains, but the reputation score r_{u_i} of u_i as well. The problem here is to find a way to achieve the following:*

1) *Keep the information of each u_i private, even if C colludes with RA (provide unlinkability);*
2) *Keep each vote on a report hidden by the linking of voters;*
3) *Add each vote to the reputation of the corresponding user so that the reputation of each user will be updated;*
4) *Prevent double voting;*
5) *Trace back a user who is acting maliciously (provide accountability);*

Threat Model: The protocol that is presented in this paper, assumes that the adversaries follow the Dolev-Yao threat model [21]. In Dolev-Yao adversarial model, adversaries can overhear and intercept any message that is exchanged between two or more parties in the network. In addition to that, they can keep history of the exchanged messages and use them later on in an attempt to learn more than what it is prescribed.

Assumptions: First of all, we assume that any number of parties on this network may collude to break the anonymity of other user(s) in the network (unbounded collusion).

Furthermore, we assume that each authority has a certificate. With this way, attacks such as denial of service can be avoided since users can authenticate authorities with which they communicate.

IV. Description of LOTS

In this section, we introduce our protocol which satisfies the above mentioned criteria and offers privacy preserving mechanisms for the users of a participatory sensing application. Before we proceed with the actual description of the protocol we provide a high-level overview of the tasks that users are able to execute through the participatory application.

A user u_i registers to the application and joins a group of users ($group_t$) that is created by RA. As we will explain later on, with the use of group signatures, u_i can prove that is a valid user without revealing her real identity. Every time that u_i wants to publish a new report, she contacts C and proves that is a member of $group_t$. C processes the request and if it is considered as valid it publishes the corresponding report to the rest of the community so that everyone can see it. The last task that a user can execute is to vote for the validity of a report or just for the importance of the published information. To this end, user sends a voting request to C. C is responsible for checking that this user has not voted before for this report and if so, proceeds by adding her vote to the corresponding report. Otherwise the request is considered as invalid and C drops the connection. In the following sections these steps are described in details.

A. Setup Phase

Installation: User u_i registers her smart-phone by installing the participatory sensing software from an application distributed market (ADM) such as Android Market or App Store. The authorization at this step could happen during the installation process as employed by the Android phones or at the start of an application as on iPhones. During installation, a credential $cred_{u_i}$ is issued and stored to user's device. The stored credential contains the following attributes[1]:

- A unique serial number S_{u_i} which is encrypted with the public key of RA
- A timestamp t_{init} which contains the date and time that u_i installed the application and retrieved S_{u_i}
- The username (usr_{market}) that u_i has registered with the ADM from which she downloaded the software (e.g username used for the App Store).

So, the credential that user receives, contains the following information:

$$\langle E_{RA}(S_{u_i}), E_{RA}(t_{init}), E_C(E_{RA}(usr_{market}))\rangle,$$

where the notation $E_x(.)$ refers to the results of the application of an encryption function with the public key of entity x.

B. User Registration

The first time that u_i connects to the service, contacts RA in order to register as a new user. By revealing some of the attributes that are contained in $cred_{u_i}$ she needs to prove that she obtained a valid credential from ADM. This can be done using the proof of knowledge described in [24].

[1]Private credentials and how attributes can be encoded into credentials have become the subject of an extensive research with many different proposed schemes and suggested approaches. For more information, we refer the reader to [22], [23].

So, u_i reveals to RA the unique serial number $E_{RA}(S_{u_i})$ that received during the installation phase as well as $E_{RA}(t_{init})$. Upon reception, RA decrypts these values with sk_{RA} and finds S_{u_i} and t_{init}. RA has a table in which stores all the hashes of the unique serial numbers that receives from newcomers. Let $H(.)$ be a secure cryptographic hash function such as $SHA3$. RA calculates $H(S_{u_i})||H(t_{init})$ and checks via a table lookup if the generated value already exists in the table. If not, she stores the generated hash to the table. Otherwise, drops the connection since the request is coming from an already registered user.

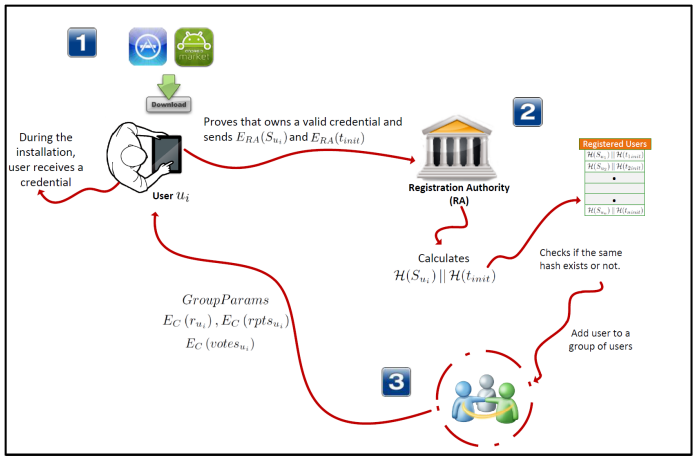

Fig. 1. Registration Phase

Group Signatures: In order for u_i to actively participate in the community she must be able to prove to the authorities that she is a valid user. In order to do so, she will have to provide some kind of evidence derived from RA and prove that she has already registered. To this end, RA sets a time interval t_{int} for which every new user that requests a registration will become a member of a group $group_t$. More precisely, u_i by joining this group will be able to anonymously sign messages on behalf of that group. Signatures can be verified with respect to a single group public key, but they do not reveal the identity of the signer. The joining procedure is implemented by a joint computation between u_i and RA in such a way that the private key of u_i remains secret. Then, RA creates a private group key on u_i and issues a membership credential so that u_i may later submit reports anonymously.

Newcomer Additional Parameters: Apart from the addition of u_i to $group_t$, RA creates and sends to u_i the following parameters:

- The reputation score r_{u_i} of u_i, encrypted with the public key pk_C of C. At the registration step the reputation score will have a negative value not only for motivating the user to participate in the community but also as a metric to prevent whitewashing attacks[2] (also known as newcomer attacks).

[2]Whitewashing attacks occur when a user abuse the system by letting her reputation degrade and then escapes the consequences by using system vulnerability to repair their reputation. Most common practice is to reenter the system with a new identity and a fresh reputation [25]. The attack is facilitated by the availability of cheap pseudonyms [26]. For a detailed description of whitewashing attacks and defensive mechanisms we refer reader to [27].

- A list ($rpts_{u_i}$) encrypted with the public key of C, that will hold a list of all the reports that u_i will publish. The initial value will contain a signature $\sigma(vr_{u_i})$ on a random nonce generated from RA.
- A list ($votes_{u_i}$) encrypted with the public key of C, that will hold a list of all the reports that u_i will vote for. In the beginning, the value of this list will contain a signature $\sigma(vr_{u_i})$ on a random nonce generated from RA.

So, at the end of the registration step RA sends back to u_i the following:

$$\langle GroupParams, E_C\left(r_{u_i}\right), E_C\left(rpts_{u_i}\right), E_C\left(votes_{u_i}\right)\rangle.$$

C. Report Submission

When u_i wants to publish a report, she needs to do it through C. Each time that u_i contacts C, uses anonymous authentication in order to prove that is a valid user. We assume that u_i wants to publish a report $R_j^{u_i}$. To achieve this, u_i uses a group signature to sign a nonce challenge that is provided by C. Then, C validates that the signed message comes from a registered user and if so, sends back to u_i a successful message. Upon reception, u_i sends back to C the following message:

$$\left\langle E_C\left(R_j^{u_i}\right), E_C\left(r_{u_i}\right), E_C\left(rpts_{u_i}\right)\right\rangle$$

C decrypts $E_C\left(R_j^{u_i}\right)$ with sk_C, creates a unique serial number ($S_{R_j^{u_i}}$) for the submitted report and adds the reputation score of u_i to it by decrypting $E_C\left(r_{u_i}\right)$. Then, all users that read $R_j^{u_i}$ will be able to see the reputation of the user who posted the corresponding report.

In addition, C has a table $T_{reports}$ where she stores all the hashes of report lists that she receives from the users. So, she calculates $H\left(E_C\left(rpts_{u_i}\right)\right)$ and checks if this value already exists in $T_{reports}$. If so, she removes it from $T_{reports}$, decrypts $E_C\left(rpts_{u_i}\right)$, adds the newly submitted report $S_{R_j^{u_i}}$ and calculates $H\left(E_C\left(rpts_{u_i}\right)\right)$ which is stored in the $T_{reports}$ table[3]. Finally, C publishes $R_j^{u_i}$ along with the reputation r_{u_i} and sends back to u_i an acknowledgment as well as the updated list of reports.

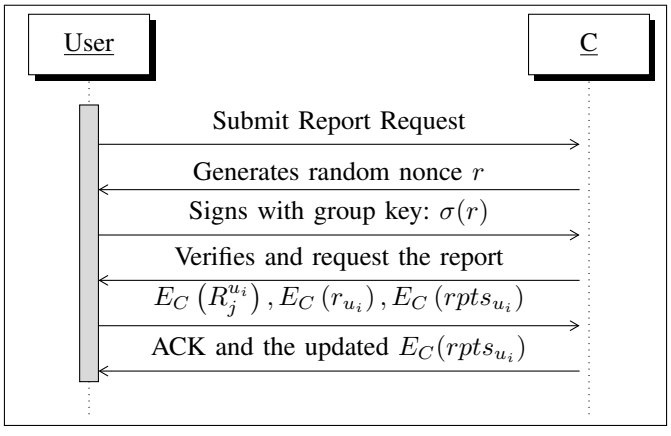

Fig. 2. Report Submission

[3]When a user has not previously published any report, C will not have a hash mapping in $T_{reports}$, can recognize the ciphertext of an empty list by contacting RA.

D. Voting

Lets assume that a user u_k wishes to vote for a report $R_j^{u_i}$. First, she proves to C that is a registered user and then sends the following:

$$\left\langle E_C\left(votes_{u_k}\right), E_C\left(S_{R_j^{u_i}}, v_k\right)\right\rangle,$$

where $votes_{u_i}$ is a list of all the reports that u_k has voted for, $S_{R_j^{u_i}}$ is the serial number of the current report that she wants to vote for and v_k is her actual vote. Upon reception, C decrypts the message and processes the submitted request as follows:

- *Case 1 – User has not submitted votes for any other report:* In that case, when C opens $E_C\left(votes_{u_k}\right)$ recognizes that it does not contain a list of reports that the user has voted for. More precisely, the content should be a signature from RA. So, C first checks a table T_{votes} to see if $E_C\left(votes_{u_k}\right)$ already exists. If not, validates that the message is signed by RA. Then, stores the received signature (σ) in T_{votes}, calculates $H(\sigma||S_{R_j^{u_i}})$ and adds it to T_{votes}. Then, updates user's voting list by setting $votes_{u_k}$ equal to $E_C\left(S_{R_j^{u_i}}, H\left(S_{R_j^{u_i}}||\sigma\right)\right)$, and sends it back to u_k. Furthermore, she updates the votes in the corresponding report.
- *Case 2 – User has submitted vote(s) for other report(s):* When C opens $E_C\left(votes_{u_k}\right)$ will see the following:
  - A list $votes_{u_k} = \left\{S_{R_1^{u_i}}, \ldots, S_{R_k^{u_i}}\right\}$ where $k \neq j$, of serial numbers that corresponds to the reports that u_k has already voted for.
  - The hash value:
    $$H' = H(S_{R_k^{u_i}}||H(S_{R_{k-1}^{u_i}}|| \ldots ||H(S_{R_1^{u_i}}||\sigma)))$$
    that is generated every time that she submits a vote for a published report.

So, C checks if $S_{R_j^{u_i}} \in votes_{u_k}$. If not, C checks if H' exists in T_{votes} and update $votes_{u_k}$ by adding a new serial number and updating the hash value $H(S_{R_j^{u_i}}||H')$. Then, she sends back to u_k the list of votes and updates the votes in the corresponding report.

E. Reputation Update

Since every user can vote for every report and the reputation of users is not actively connected to the votes of their reports, we must design a mechanism in which the reputation of users and the reputation of the corresponding reports will be updated frequently. In order to achieve that, we assume that there is a time interval t_{rep} in which all users have to contact C in order to update their reputations. So, each user u_i will send $rpts_{u_i}$ and r_{u_i} encrypted with pk_C. Upon reception, C decrypts $rpts_{u_i}$ with sk_C and finds a list with all the reports that u_i has published. Then, calculates the hashes and checks its freshness. If C finds any old $rpts_{u_i}$ changes the corresponding reputation to the initial state (negative reputation). For the rest of the lists, that are considered valid, C finds the current reputation scores for the corresponding reports, calculates the average, encrypts it with pk_C and updates r_{u_i}. In addition, C adds a timestamp in r_{u_i} that will prove when was the last time that a user updated her reputation. Then, C sends back to u_i the updated reputation.

V. Security Analysis

In this section, the behavior of $LotS$ when different types of attacks take place is analyzed. In all the attacking scenarios, it is assumed that the authorities RA and C follow the protocol specifications but they can keep history of the exchanged messages in an attempt to learn more than what has been prescribed. Additionally, attackers can collude in order to find the real identity of the users and, thus, break their privacy.

Breaking the Anonymity of u_i: As described in Section IV the first step for a user before start using the service is to register through RA. In this phase, u_i first contacts RA and proves that she owns a valid credential and a unique serial number from ADM. Then, RA adds u_i to a group ($group_t$) of users that share the same group signature key. Now, every time that u_i wishes to publish a new report or vote for a published report she signs a message on behalf of that group. As a consequence, the authorities will be able to know only the group in which u_i belongs to, but none of them will be able to find the exact user that published the report[4]. The only case where an authority can extract the real identity of a user is in the extreme case where at the time that u_i tries to publish a new report or tries vote for a published report, is the only member of $group_t$. But still, if this extreme case is considered, RA needs to collude with ADM in order to find the real identity of u_i through the unique serial number S_{u_i} that ADM assigned to u_i and is also known to RA from the registration phase. This problem can be easily solved; if instead of ADM storing $\{S_{u_i}\}$ at a user's device, she will blindly sign a random number that will be sent from u_i. Then u_i will have to prove to RA that she owns a valid signature from ADM. So, even if RA colludes with ADM it will not achieve to reveal the real identity of the user.

Double Voting: Assume that a user u_i votes for a report $S_{R_{u_j}}$. The serial number of the report is added to user's voting list $votes_{u_i}$. Suppose that u_i tries to vote again for the same report. In the simple case where u_i contacts C with the latest list $votes_{u_i}$, C will immediately realize that $S_{R_{u_j}}$ already exists in $votes_{u_i}$ so it will drop the connection. In order for u_i to avoid this detection, she can send to C a list $votes'_{u_i}$ that does not contain $S_{R_{u_j}}$ (i.e a list that was received in the past). Then, C will check for the hash value $H(.)$ that is contained in $votes'_{u_i}$ and she will notice that the received value does not exist in the table T_{votes}. So, C will realize that u_i has sent a non-valid list and will not process the request.

Using the Reputation of Another User: Suppose that a user u_j wishes to publish a report but instead of using her reputation, tries to use the reputation of another user u_i. In order to achieve it, u_j must gain access to $rpts_{u_i}$. Since all the communication between the users and the authorities is encrypted the only method for u_j to get $rpts_{u_i}$ is either to directly cooperate with u_i or to collude with C. For the first case, it is clear that the consent of the user u_i is needed. The privacy of u_i can be breached with her own consent. In addition to that, u_i will need to receive from u_j the updated $rpts_{u_i}$ that it was generated and sent to u_j after publishing the new report. Otherwise, u_i will not be able to publish a new report again. Hence, if u_i does not update $rpts_{u_i}$ her account will be almost disabled since she will only be able to vote for a published report while her publish rights will be disabled. Regarding the second case, u_j will need to receive $rpts_{u_i}$ from C, which have stored the history of the exchanged messages. This implies that the authorities deviate from the protocol flaw which contradicts our hypothesis.

Accountability (traceability): Anonymity and accountability are supposedly opposing factions in a zero-sum game between privacy and security. Generally speaking, we can say that an action is accountable if it can be attributed to a certain entity. Even though anonymous communication provides enormous public benefits, it has been reportedly used to aid people to perform illegal transactions. Accountability mechanisms, allows to reveal the identity of a user who acts maliciously. As described in [20], in case of a user's misbehavior, the signature can be "opened" (with or without the help of the group members), in order to reveal the identity of the signer. In our protocol, the identity of a misbehaving user can only be revealed if at least k users from the same group cooperate. Therefore, in order to ensure the k-anonymity of the signing users [28] we assume that each group consists of at least k users. By doing this, we ensure that not only RA will be able to reveal the anonymity of a user; thus, providing a better privacy since if we assume that RA is compromised it will need the cooperation of other members of a group in order to reveal the identity of a user.

Disadvantages: Even though we have managed to provide reliable anonymity in the sense that each user keeps her real identity hidden during her transactions in the community, protecting the privacy of a user is a more complex subject. More precisely, LotS fails to prevent the profiling of a user. Recall that every time that a user wishes to publish a new report or wants to give a vote, she needs to interact with C and reveal a list with all the reports that she has published or she has voted for. So, in case C is acting maliciously, can collect data and create user profiles.

VI. Conclusions

In this paper we presented $LotS$. A privacy preserving reputation system for participatory sensing applications. $LotS$ manages to provide accountable anonymity. Users are able to exchange information in a lawful manner without being tracked. On the other hand, users who are misbehaving will loose their anonymity and they will encounter the repercussions that are defined from the community regarding malicious behaviors. Additionally, users have a reputation score for which they cannot lie about or shed while at the same time the reputation is updated and demonstrated in such way that does not compromise anonymity. In particular, $LotS$ maintains unlinkability between the identity of users and the reports they publish. Moreover, on our theoretical analysis we prove the resistance of the protocol under different kinds of attacks. As a next step we are planning to further improve our protocol by providing mechanisms that will avoid the profiling of a user. Finally, we intend to implement our protocol in actual smart-phone devices with different computational resources.

[4]The security of group signatures under the random oracle model is based on the intractability of discrete logarithm.

REFERENCES

[1] Cisco and its affiliates, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016," tech. rep., Cisco, 02 2012.

[2] I. Krontiris, F. C. Freiling, and T. Dimitriou, "Location privacy in urban sensing networks: research challenges and directions," *Wireless Commun.*, vol. 17, pp. 30–35, Oct. 2010.

[3] L. Deng and L. P. Cox, "Livecompare: grocery bargain hunting through participatory sensing," in *Proceedings of the 10th workshop on Mobile Computing Systems and Applications*, HotMobile '09, (New York, NY, USA), pp. 4:1–4:6, ACM, 2009.

[4] G.-S. Ahn, M. Musolesi, H. Lu, R. Olfati-Saber, and A. T. Campbell, "Metrotrack: predictive tracking of mobile events using mobile phones," in *Proceedings of the 6th IEEE international conference on Distributed Computing in Sensor Systems*, DCOSS'10, (Berlin, Heidelberg), pp. 230–243, Springer-Verlag, 2010.

[5] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, "Bikenet: A mobile sensing system for cyclist experience mapping," *ACM Trans. Sen. Netw.*, vol. 6, pp. 6:1–6:39, Jan. 2010.

[6] N. Maisonneuve, M. Stevens, and B. Ochab, "Participatory noise pollution monitoring using mobile phones," *Info. Pol.*, vol. 15, pp. 51–71, Apr. 2010.

[7] E. Kanjo, J. Bacon, D. Roberts, and P. Landshoff, "Mobsens: Making smart phones smarter," *IEEE Pervasive Computing*, vol. 8, pp. 50–57, Oct. 2009.

[8] 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 *Proceedings of the 6th ACM conference on Embedded network sensor systems*, SenSys '08, (New York, NY, USA), pp. 337–350, ACM, 2008.

[9] K. Shilton, "Four billion little brothers?: privacy, mobile phones, and ubiquitous data collection," *Commun. ACM*, vol. 52, pp. 48–53, Nov. 2009.

[10] S. B. Eisenman and A. T. Campbell, "Skiscape sensing," in *Proceedings of the 4th international conference on Embedded networked sensor systems*, SenSys '06, (New York, NY, USA), pp. 401–402, ACM, 2006.

[11] E. P. Stuntebeck, J. S. Davis, II, G. D. Abowd, and M. Blount, "Healthsense: classification of health-related sensor data through user-assisted machine learning," in *Proceedings of the 9th workshop on Mobile computing systems and applications*, HotMobile '08, (New York, NY, USA), pp. 1–5, ACM, 2008.

[12] 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 *Proceedings of the 7th international conference on Mobile systems, applications, and services*, MobiSys '09, (New York, NY, USA), pp. 55–68, ACM, 2009.

[13] Y. F. Dong, S. Kanhere, C. T. Chou, and N. Bulusu, "Automatic collection of fuel prices from a network of mobile cameras," in *Proceedings of the 4th IEEE international conference on Distributed Computing in Sensor Systems*, DCOSS '08, (Berlin, Heidelberg), pp. 140–156, Springer-Verlag, 2008.

[14] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "AnonySense: Privacy-aware people-centric sensing," in *Proceedings of the 2008 International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pp. 211–224, ACM Press, June 2008.

[15] O. Berthold, H. Federrath, and S. Köpsell, "Web mixes: a system for anonymous and unobservable internet access," in *International workshop on Designing privacy enhancing technologies: design issues in anonymity and unobservability*, (New York, NY, USA), pp. 115–129, Springer-Verlag New York, Inc., 2001.

[16] E. De Cristofaro and C. Soriente, "PEPSI—privacy-enhanced participatory sensing infrastructure," in *Proceedings of the fourth ACM conference on Wireless network security*, WiSec '11, (New York, NY, USA), pp. 23–28, ACM, 2011.

[17] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," *SIAM J. of Computing*, vol. 32, no. 3, pp. 586–615, 2003. extended abstract in Crypto'01.

[18] I. Boutsis and V. Kalogeraki, "Privacy preservation for participatory sensing data," in *Pervasive Computing and Communications (PerCom), 2013 IEEE International Conference on*, pp. 103–113, 2013.

[19] D. Christin, C. Rosskopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, "Incognisense: An anonymity-preserving reputation framework for participatory sensing applications.," in *PerCom* (S. Giordano, M. Langheinrich, and A. Schmidt, eds.), pp. 135–143, IEEE, 2012.

[20] D. Chaum and E. Van Heyst, "Group signatures," in *Proceedings of the 10th Annual International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'91, (Berlin, Heidelberg), pp. 257–265, Springer-Verlag, 1991.

[21] D. Dolev and A. C.-C. Yao, "On the security of public key protocols.," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–207, 1983.

[22] D. Chaum and J.-H. Evertse, "A secure and privacy-protecting protocol for transmitting personal information between organizations," in *Proceedings on Advances in cryptology—CRYPTO '86*, (London, UK, UK), pp. 118–167, Springer-Verlag, 1987.

[23] J. Camenisch and A. Lysyanskaya, "An efficient system for non-transferable anonymous credentials with optional anonymity revocation," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT '01, (London, UK, UK), pp. 93–118, Springer-Verlag, 2001.

[24] J. Camenisch and T. Groß, "Efficient attributes for anonymous credentials," *ACM Trans. Inf. Syst. Secur.*, vol. 15, pp. 4:1–4:30, Mar. 2012.

[25] K. Lai, M. Feldman, I. Stoica, and J. Chuang, "Incentives for cooperation in peer-to-peer networks," 2003.

[26] E. Friedman and P. Resnick, "The social cost of cheap pseudonyms," 1998.

[27] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Comput. Surv.*, vol. 42, pp. 1:1–1:31, Dec. 2009.

[28] L. Sweeney, "K-anonymity: A model for protecting privacy," *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, vol. 10, pp. 557–570, Oct. 2002.