Location privacy-preserving mobile crowd sensing with anonymous reputation

Yi, Xun; Lam, Kwok-Yan; Bertino, Elisa; Rao, Fang-Yu

2019

Yi, X., Lam, K.-Y., Bertino, E., & Rao, F.-Y. (2019). Location privacy-preserving mobile crowd sensing with anonymous reputation. *Computer Security – ESORICS 2019*, 387-411.
doi:10.1007/978-3-030-29962-0_19

<https://hdl.handle.net/10356/138424>

https://doi.org/10.1007/978-3-030-29962-0_19

© 2019 Springer Nature Switzerland AG. All rights reserved. This paper was published in *Computer Security – ESORICS 2019* and is made available with permission of Springer Nature Switzerland AG.

Downloaded on 26 Apr 2023 22:26:05 SGT

Location Privacy-Preserving Mobile Crowd Sensing with Anonymous Reputation

Xun Yi¹(B), Kwok-Yan Lam², Elisa Bertino³, and Fang-Yu Rao³

¹

RMIT University, Melbourne, Australia xun.yi@rmit.edu.au

²

Nanyang Technological University, Jurong West, Singapore kwokyan.lam@ntu.edu.sg

³

Purdue University, West Lafayette, USA bertino@purdue.edu

Abstract. In this paper, we give a location privacy-preserving solution for the mobile crowd sensing (MCS) system. The solution makes use of the blind signature technique for anonymous authentication and allows a mobile user to participate in the MCS for certain times set in the registration. Furthermore, we introduce a concept of anonymous reputation for mobile users on the basis of the blind signature technique as well. An anonymous reputation can be referred by the MCS platform when assigning tasks to a mobile user and can be upgraded or downgraded by the MCS platform, depending on the quality of reports submitted by the mobile user. For the security analysis, we provide security proofs for our solution on the basis of our formal definitions for anonymity, unlinkability and unforgeability for MCS. The performance analysis and experiments have shown that our solution is more efficient than existing solutions for MCS based on the blind signature technique.

Keywords: Mobile crowd sensing • Location privacy protection •

Anonymity • Blind signature • Reputation

Introduction

An emerging category of devices at the edge of the Internet are consumer-centric mobile sensing and computing devices, such as smartphones, music players, and in-vehicle sensors. These devices will fuel the evolution of the Internet of Things as they feed sensor data to the Internet at a societal scale [14]. Mobile crowd sensing (MCS) has been gaining popularity, with several systems and applications being proposed to leverage users' mobile devices to measure environmental context. The building blocks of an MCS are the central authority that provides the application (the platform) and the participants (the users) that contribute their collected data. Applications of MCS include monitoring city noise, city climate, people density, emergency behavior, traffic anomalies and even detecting earthquakes [15,16].

One of the most significant advantages of these MCS applications is that given the large number of existing cellular users, they have the potential to collect data like never before and from places not economically feasible before, and in a fast, easy, and cost-effective manner. For example, traffic congestion applications with MCS have the potential to collect real-time data not only from main interstate roads but also from secondary and even tertiary roads, something that is very costly using current technologies. Deploying static sensors over all roads will be economically expensive in terms of capital, installation and maintenance costs.

The MCS data is usually tagged with locations of mobile users. Mobile users' locations are vulnerable to malicious attacks. Even if mobile users are protected by fake identities like pseudonym, an adversary can utilize their locations to infer the private information, such as political affiliations, alternative lifestyles, or medical problems. Therefore, it is especially essential to achieve location privacy protection.

Location privacy has been vastly studied in Location Based Service (LBS) and crowd sensing system, where the servers or platforms are often regarded untrusted. To preserve location privacy, various methods were proposed to prevent the servers or platforms inferring users' exact locations. Some surveys on location privacy-preserving mechanisms for MCS have been done in [10,31]. In terms of the underlying techniques for building MCS with location privacy protection, current solutions can be classified into: k -anonymity based, blind signature based, and group signature based.

In the existing solutions for location privacy protection in MCS, there are some problems, summarized as follows:

- Trusted third party problem: k -anonymity based solutions [30,32] assume the existence of a trusted third party (TTP) between mobile users and the platform. The TTP knows the exact locations of mobile users and performs location data perturbation before forwarding it to the platform. It is hard to ensure the TTP security against hackers.
- Linkability problem: Blind signature based solution [21] allows the platform to trace mobile users with the same certified pseudonym. In some applications, it may not be acceptable for privacy concern. The problem can be overcome by using multiple blind signatures, each of them can be used once only. But a mobile user has to keep many blind signatures. Ramzan and Ruhl [26] suggested a method to reduce the number of blind signatures.
- Secret sharing problem: In the solutions [20,28] based on group signature, a mobile user is issued by the platform a kind of secret to prove to the platform the eligibility for participation in MCS. A dishonest mobile user may share the secret with others. Although this problem also exists in blind signature based solutions, it is not serious because a blind signature can be used once only.

The above problems are interdependent. It is a challenge to provide a solution for MCS to overcome all the problems.

Besides the location privacy issue, recent research in MCS finds another important challenge: the need to understand user intentions and to quantify their reputation. A user might simply think that an MCS application is not good for the community

and protest it by contributing incorrect data intentionally or might have a faulty mobile device that takes imprecise or wrong measurements. Trustworthiness of the collected data is a primary concern for both the platform and the end users who request sensed data as a service [2].

Current research on the reputation of mobile user has not considered the location privacy for mobile users. To protect the location privacy of mobile users in MCS, the mobile users have to perform anonymous authentication with the MCS platform. Due to the anonymity requirement, it is difficult for the MCS platform to score user reputation. It is a challenge to make use of user reputation in MCS with user location privacy and unlinkability.

Our Contribution. In this paper, we propose a new solution to protect location privacy for mobile users in MCS. Our solution is built on the blind signature technique.

The basic idea for anonymous authentication is as follows: At first, the platform issues a mobile user a blind signature on the times that the user can participate in MCS, so called anonymous certificate. When the mobile user wishes to participate in MCS, he submits the anonymous certificate to the platform for authentication and then the platform returns him a new anonymous certificate, a blind signature on the remaining times that the user can participate in MCS. During the authentication, the platform and the user agree on a secret key to protect information exchanged later, such tasks, reports, etc. After the user submits the reports, the platform rewards the user with e-cash. The user can spend the e-cash anonymously.

In addition, we introduce a new concept of anonymous reputation to improve MCS performance. The basic idea for anonymous reputation is similar to the basic idea for anonymous authentication. Each time when the mobile user submits the report, the platform issues a blind signature on the level of the user reputation, so-called anonymous reputation, on the basis of the quality of the report. Next, when the user wishes to participate in MCS, he submits his anonymous reputation to the platform for reference. The platform usually assigns a task to the user with the highest level of anonymous reputation. Our solution has the following security properties:

- Anonymity. The MCS platform cannot determine any mobile user's identity when the user participates in MCS.
- Unlinkability. The MCS platform cannot determine whether two anonymous certificates belong to the same mobile user.
- Unforgeability. A mobile user cannot forge any anonymous certificate or any anonymous reputation to obtain more than the specified number of MCS accesses or a reputation higher than the specified level. Even if several mobile users collude (by sharing their anonymous certificates and anonymous reputation) they cannot obtain more MCS accesses than they are paid for as a group or higher level of reputation than they are awarded for as a group.

In addition, our solution has the following system properties:

- Limited Access: The number of MCS accesses is limited. That is, the mobile user is allowed to access the MCS system only a specified number of times. After that his access privileges (implicitly, i.e. without any communication among the platform and users) terminate.
- Unshareable: It is impossible to share a single anonymous certificate or a single anonymous reputation among two or more mobile users.

Our basic idea for anonymous authentication is motivated by [26], where two solutions were proposed. In the first solution, a user needs to keep a number of blind signatures, which is the bit length of the times that the user can access the system, and the user keeps track of the remaining subscription length, and shows only part of that information to the supplier when accessing the service. The second solution adopts signatures of knowledge [6] that enable a user to prove to the server that user is in possession of a valid RSA blind signature of the server on the user's public key. Although a user only has to keep a constant number secrets, i.e., his secret key as well as the blind signature, the invocation of the protocol proving the knowledge of both his secret key and the blind signature incurs a lot of computational overhead. However, in our solution, a user needs to keep one blind signature only and submits one blind signature to the platform for authentication. Therefore, in terms of anonymous authentication, our solution is more efficient than [26].

In addition, in our solution, for authentication, a mobile user submits an anonymous certificate to the platform and obtains a new anonymous certificate from the platform. This idea is similar to [3]. But [3] requires to run the zero-knowledge proof between the user and the server for authentication, while our solution needs to verify a blind signature only. Thus, in terms of anonymous authentication, our solution is also more efficient than [3].

Organization. The rest of our paper is organized as follows. We survey the related works in Sect.2. Then we give the security definitions in Sect.3, describe the proposed solution in Sect.4, and analyze the security and performance for the proposed protocols in Sect.5. Conclusions are drawn in the last section.

Related Works

A popular approach to preserving privacy of user's data is anonymization [29], which removes any identifying information from the sensor data before sharing it with a third party. k -anonymity is a property possessed by certain anonymized data. A release of data is said to have the k -anonymity property if the information for each person contained in the release cannot be distinguished from at least $k-1$ individuals whose information also appear in the release. Regarding k -anonymity of location privacy, at least k users' locations are mixed into a group, in which an adversary cannot distinguish one user's location from the rest of others'. In the scenario of MCS system, the MCS server may be malicious, thus directly exposing users' raw locations is harmful to the location privacy. A trusted third party, which

is a cellular service provider, is supposed to protect the location privacy and process the sensory data.

In 2014, To et al. [30] proposed a framework for protecting privacy of worker locations in MCS. In their framework, every worker subscribes to a cellular service provider (CSP) that has access to the worker locations. The CSP sanitizes the worker locations dataset using the powerful differential privacy model [13]. First, workers send their locations to the trusted CSP which builds and releases a Private Spatial Decomposition (PSD) according to a privacy budget mutually agreed upon with the workers. A PSD [11] is a spatial index transformed according to differential privacy, where each index node is obtained by releasing a noisy count of the data points enclosed by that node's extent. When the MCS server has a task t , it queries the PSD to determine a geocast region (GR) that encloses with high probability workers in relative proximity to t . Next, the MCS server initiates a geocast communication [23] process to disseminate t to all workers within the GR. In 2017, Wang et al. [32] considered how to incentivize mobile users to participate in MCS while preserving their location privacy. To reduce the risk of location privacy disclosure, they utilized k -anonymity. Their basic idea is: (1) Users report their locations to a trusted third party for location privacy protection. Along with those locations, the users will also claim costs for taking the sensing tasks; (2) The trusted third party is supposed to perform aggregation on the locations and interact with an untrusted crowd sensing platform. In view of k -anonymity privacy, the trusted third party constructs groups with each group size no less than k . Based on the group aggregation results, group values and group costs will be computed; (3) According to group values and costs, the crowd sensing platform selects the winning groups and calculate corresponding group payments; (4) The users in winning groups are winning users, and their payments will be computed based on group payments; (5) Winning users will undertake the sensing tasks, and the trusted third party will process the sensory results within the same groups, such as computing the mean values; (6) The processed sensory data tagged with groups centroids will be uploaded to the crowd sensing platform; (7) The crowd sensing platform pays the winning users the determined values through the trusted third party according to (3) and (4). k -anonymity based MCS is efficient, however, it assumes the existence of a trusted third party which knows the exact locations of mobile users. It may be hard to ensure a trusted third party security against hackers.

The second approach to preserving location privacy in MCS is based on blind signature [8]. A blind signature is a form of digital signature in which the content of a message is disguised (blinded) before it is signed. The resulting blind signature can be publicly verified against the original, unblinded message in the manner of a regular digital signature.

In 2013, Konidala et al. [21] proposed the anonymous authentication of visitors protocol to authenticate the information reported by a visitor inside a thematic park without divulging the visitor's identity. The protocol protects the visitors' location privacy with the partially blind signature [1]. A partially blind signature here has two portions: one portion consists of the message that is hidden by the user (e.g., demographic details, such as age, gender, nationality, height, dietary

restrictions and other health issues, as well as preferences for rides; must go and must skip attractions and etc.) and in the other portion, the signer can explicitly embed necessary information such as issuing date, expiry date, signer's identity and etc.). The protocol is executed in two phases: (1) certified pseudonym issuing phase, and (2) subsequent interaction phase. In the first phase, the visitor generates a pseudonym P and utilizes a partially blind signature to hide P in a blinded message B , which is sent to the park operator. The operator inputs an expiry date while digitally signing B . In the end of this phase, the visitor derives a certified pseudonym (i.e., blind signature). In the second phase, before the expiry date, the visitor can send the certified pseudonym to the operator repeatedly to receive the optimal route, and the dynamically calibrated personalized time slots for various attractions in the park, as well as rewards.

Because the operator has no clue about visitor's pseudonym in the first phase, he cannot link the pseudonym to the visitor in the second phase. But the operator is able to trace the visitor with the same pseudonym. For some applications, the linkability may not be allowed.

The third approach for location privacy protection in MCS is based on group signature. A group signature scheme is a method for allowing a member of a group to anonymously sign a message on behalf of the group. The concept was first introduced by David Chaum and Eugene van Heyst in 1991 [9]. For example, a group signature scheme could be used by an employee of a large company where it is sufficient for a verifier to know a message was signed by an employee, but not which particular employee signed it.

AnonySense [20,28] is an MCS system based on group signature. The system periodically posts the tasking campaign and when the participants are in public locations, locations considered by the participant as nonsensitive ones, they download all available tasks from a tasking service. For each connection, the participant performs an anonymous authentication, based on a group signature defined by direct anonymous attestation [5], in order to prove to the system that it is a valid participant, but without revealing its identity. Therefore, the system only learns that some participants are in public locations but nothing else, which is the main advantage of this system. However, its main drawback is that, since the system only learns that some users are in public locations, the system cannot predict how many users are likely to visit a particular region and guarantee a good inference and data analysis. In addition, some users may share their secret key with others, which results that unauthorized users can also participate in MCS without registration.

On the reputation of mobile users in MCS, Kantarci et al. [18] proposed data trustworthiness assurance in user incentivization using statistical-based and recommendation - based user reputation-awareness methods, and a Social Network-Assisted Trustworthiness Assurance (SONATA) [19], which is a recommendation-based approach to identify malicious users who manipulate sensor readings to spread disinformation. SONATA adopts a vote-based trustworthiness analysis to minimize the manipulation probability in an MCS framework. Pouryazdan et al. [24] introduced anchor nodes, which are deployed as trusted entities in an MCS system in order to improve the platform and user

utility by eliminating adversaries at the end of a recommendation-based user recruitment process. Pouryazdan et al. [25] also introduce a new metric - collaborative reputation scores. Ren et al. [27] proposed a participant selection method to choose well-suited users for assigning tasks as well as to consider a reputation management scheme to evaluate the trustworthiness of the contributed data. These works on the reputation of mobile user have not considered the location privacy.

Security Model

In this section, we define the security of location privacy-preserving (LPP) protocol for MCS with anonymous reputation as follows.

Participants, Initialization, Registration. An LPP protocol involves two kinds of protocol participants: (1) An MCS platform P , which provides MCS service, authenticates mobile users, assigns tasks to mobile users on the basis of their reputation, receives reports from the mobile users, and pays the mobile users for their reports and scores their reputation. (2) A group of mobile users $U_1, U_2, \dots, U_n$, who participates in MCS, receives tasks from the platform, submits reports to the platforms, and receives payment from the platform.

Prior to any execution of the protocol, we assume that an initialization phase occurs. During initialization, the platform generates public parameters for the protocol, which are available to all participants.

We assume that each mobile user U_i runs a registration protocol, with the MCS platform P . During registration, the MCS platform issues an anonymous certificate (i.e., a blind signature) and an anonymous reputation (i.e., a blind signature) at level 1 to the user.

Execution of the Protocol. After registration, a mobile user U_i can run the LPP protocol with the MCS platform P . The protocol includes three phases as follows.

- Authentication: When a mobile user wishes to participate in MCS, he submitshis anonymous certificate and location to the MCS platform for authentication. The platform checks if the anonymous certificate has been used before. If not, the mobile user and the MCS platform agree on a secret key to protect information exchanged later.

Remark: Like a blind signature, each anonymous certificate can be used once only. We assume that the MCS platform keeps all used anonymous certificates in its database.

- Task assignment: The platform sends a list of tasks (near the location ofthe user) with costs to the mobile user. The user chooses a task and submits his anonymous reputation to the platform if any. The platform check

if the anonymous reputation has been used before. If there are several users to compete for one task, the platform assigns the task to the user with the highest level of reputation. The information exchanged in this phase is protected with a secret key agreed on in the authentication phase.

Remark: Like an anonymous certificate, each anonymous reputation can be used once only. We assume that the MCS platform keeps all used anonymous reputation in its database.

- **Report and Reward:** The mobile user performs the task at the specified location and sends a report back to the MCS platform. The report is protected with a secret key agreed on in the authentication phase. After checking the report, the MCS platform rewards the mobile user with bitcoins. In addition, the platform upgrades or downgrades the reputation of the user accordingly.

Now we define the anonymity for mobile users. To define the anonymity for mobile users, we assume that the MCS platform is malicious and attempts to reveal the identity of a mobile user in the protocol.

We use a game to formally define the anonymity. In the game, suppose that the adversary A (i.e., the MCS platform) chooses two mobile users U_0 and U_1 from all users, and the challenger C randomly chooses a bit b and runs the LPP protocol with the adversary A on behalf of the mobile user U_b . As defined in the LPP protocol, the adversary A is able to view all messages exchanged with the mobile user U_b , including authentication, task assignment, report and reward messages from the mobile user U_b . In the end, the adversary outputs a bit b' (i.e., his guess about the bit b chosen by the challenger).

The adversary A wins the game if $b' = b$. The probability of the adversary A winning the game is called the advantage of the adversary A in attacking the anonymity of the LPP protocol, denoted as $\text{AnonymityAdv}_A^{\text{LPP}}(k)$, where k is a security parameter.

Definition 1. An LPP protocol has anonymity for mobile users if for any Probabilistic Polynomial Time (PPT) adversary A , there is a negligible function $\epsilon(\cdot)$, such that $|\text{AnonymityAdv}_A^{\text{LPP}}(k) - 1/2| < \epsilon(k)$, where k is a security parameter.

Next we define the unlinkability for the mobile user. We also assume that the MCS platform is malicious and attempts to link two accesses from the same mobile user. We use a game to formally define the unlinkability. In the game, suppose that the adversary A (i.e., the MCS platform) chooses two mobile users U_0 and U_1 from all users, and the challenger C runs the LPP protocol with the adversary A on behalf of the mobile user U_0 and U_1 , respectively, and also discloses the corresponding identities of the mobile users communicating with the adversary A . Next, the challenger C randomly chooses a bit b and runs the LPP protocol with the adversary A on behalf of the mobile user U_b without telling b to the adversary. As defined in the LPP protocol, the adversary A is able to view all messages

exchanged with the mobile user U_b . In the end, the adversary outputs a bit b (i.e., his guess about the bit b chosen by the challenger).

The adversary A wins the game if $b' = b$. The probability of the adversary A winning the game is called the advantage of the adversary A in attacking the unlinkability of the LPP protocol, denoted as $\text{UnlinkabilityAdv}^{LPP}(k)$, where k is a security parameter.

Definition 2. An LPP protocol has unlinkability for mobile users if for any PPT adversary A , there is a negligible function $\epsilon(\cdot)$, such that $|\text{UnlinkabilityAdv}_A^{LPP}(k) - 1/2| < \epsilon(k)$, where k is a security parameter.

At last, we define the unforgeability for the MCS platform. To define the unforgeability, we assume that a group of mobile users are malicious and attempt to forge an anonymous certificate or an anonymous reputation to obtain more than the specified number of MCS accesses or a reputation higher than the specified level.

We also use a game to formally define the unforgeability. In the game, suppose that the adversary A (i.e., the group of malicious mobile users) is provided with t anonymous certificates or anonymous reputations. If the adversary A can forge any new anonymous certificate or anonymous reputation, the adversary wins the game. The probability of the adversary A winning the game is called the advantage of the adversary A attacking the unforgeability of the LPP protocol, denoted as $\text{UnforgeabilityAdv}^{LPP}(k)$.

A

Definition 3. An LPP protocol has unforgeability for the MCS platform if for any PPT adversary A , there is negligible function $\epsilon(\cdot)$, such that $\text{UnforgeabilityAdv}_A^{LPP} < \epsilon(k)$, where k is a security parameter.

Location Privacy-Preserving Protocol for MCS with Anonymous Reputations

In this section, we describe the initialization (parameter generation) and registration of the LPP protocol at first and then three phases (authentication, task assignment, report and reward) of the protocol.

Initialization

The initialization performs parameter generation as follows: Given a security parameter $k \in \mathbb{Z}$, the MCS platform chooses two large primes $p = 2p' + 1$ and

$q = 2q' + 1$, where $p, q > k$ and (p', q') are primes as well, and computes $N = pq$ and $\phi(N) = (p - 1)(q - 1) = 4p'q'$. The MCS platform chooses a public key e_a for anonymous certificate and a public key e_r for anonymous reputation, and computes the corresponding private keys d_a and d_r , such that $\gcd(e_a, \phi(N)) = 1, \gcd(e_r, \phi(N)) = 1, e_a d_a = 1 \pmod{\phi(N)}, e_r d_r = 1 \pmod{\phi(N)}$. The MCS platform publishes the public key e_a, e_r and N , but keeps the private keys d_a, d_r and p, q secret. In addition, the MCS platform publishes a secure hash function

$H : Z \rightarrow Z_N$.

Registration

Suppose that the mobile user U_i wants to access the MCS system for n times by registration, the user randomly chooses integers r, m, R, M from Z_N^* and computes

$$\begin{aligned} A_a &= r^{(2n+1)e_a} \cdot H(MCS, m) \pmod{N} \\ (1) \quad A_r &= R^{3e_r} \cdot H(MCS, M) \pmod{N} \end{aligned} \quad (2)$$

Then the user sends a registration request including $\{(n, A_a), (1, A_r)\}$ to the MCS platform.

After receiving the registration request, the MCS platform signs the blinded message by computing

$$B_a = A_a^{(2n+1)^{-1}d_a} \pmod{N} = r \cdot H(MCS, m)^{(2n+1)^{-1}d_a} \pmod{N} \quad (3)$$

$$B_r = A_r^{3^{-1}d_r} \pmod{N} = R \cdot H(MCS, M)^{3^{-1}d_r} \pmod{N} \quad (4)$$

and returns B_a, B_r to the mobile user. Note the probability of $\gcd(2n + 1, \phi(N)) \neq 1$ or $\gcd(3, \phi(N)) \neq 1$ can be ignored because $\phi(N) = 4p'q'$ and p', q' are large primes, and therefore the MCS platform can compute $(2n + 1)^{-1} \pmod{\phi(N)}$ and $3^{-1} \pmod{\phi(N)}$.

Finally, the user removes the blindness by computing

$$C_a = r^{-1} B_a \pmod{N} = H(MCS, m)^{(2n+1)^{-1}d_a} \pmod{N} \quad (5)$$

$$C_r = R^{-1} B_r \pmod{N} = H(MCS, M)^{3^{-1}d_r} \pmod{N} \quad (6)$$

In the end of registration, the mobile user obtains an anonymous certificate, $\{n, m, C_a\}$, by which the mobile user can participate in MCS for n times, and an anonymous reputation at level 1, $\{1, M, C_r\}$.

Remark. Unless otherwise specified, the user will always verify the blind signature produced by the MCS platform in our protocol.

Protocol Execution

When a mobile user U_i wishes to participate in MCS to perform any tasks near his location, he initializes a request to run the LPP protocol with the MCS platform and the protocol runs in three phases as follows:

Authentication. Assume that the mobile user U_i has an anonymous certificate $\{\ell, m, C_a\}$, by which the user can participate in MCS for ℓ times, where $\ell \geq 1$.

The mobile user U_i randomly chooses integers r', m', s from Z_N^* and computes

$$D = s^{e_a} \pmod{N} \quad (7)$$

$$k = H(s) \quad (8)$$

$$A'_a = r'^{(2\ell-1)(2\ell+1)e_a} H(MCS, m') \pmod{N} \quad (9)$$

Next, the mobile user encrypts $\{MCS, (\ell, m, C_a), A'_a\}$ with a secret key encryption algorithm, e.g., AES [12], by the secret key k . The encryption result is denoted as $E_k(MCS, (\ell, m, C_a), A'_a)$.

Then, the user submits an authentication request $\{MCS, D, E_k(MCS, (\ell, m, C_a), A'_a)\}$ to a nearby base station for mobile communication. The base

station forwards the authentication request together with the location L of the base station to the MCS platform.

Remark. We assume that the mobile user is able to submit his authentication request to the base station anonymously. For example, the mobile user may use a dual SIM smart phone¹. One SIM is for normal use and another SIM is specially developed for the purpose of MCS. The MCS SIM submits the authentication request to the base station without any additional information by which the base station can identify the mobile user.

After receiving the message $\{MCS, D, E_k(MCS, (\ell, m, C_a), A'_a), L\}$, the MCS platform computes

$$s = D^{d_a} \pmod{N} \quad (10)$$

and $k = H(s)$ and decrypt $E_k(MCS, (\ell, m, C_a), A'_a)$ with the secret key k to obtain $\{\ell, m, C_a\}$. Then the MCS platform checks if $\{\ell, m, C_a\}$ appears in the used anonymous certificate database and if

$$C_a^{(2\ell+1)e_a} = H(MCS, m) \pmod{N}. \quad (11)$$

If $\{\ell, m, C_a\}$ does not appear in the used anonymous certificate database and the Eq.(11) holds, the mobile user is authentic and the MCS platform adds $\{\ell, m, C_a\}$ into the used anonymous certificate database. In addition, the MCS platform also

¹ https://en.wikipedia.org/wiki/Dual_SIM.

keeps $\{D, k, (\ell, m, C_a), A'_a, L\}$ for next two phases of communication with the user U_i . The protocol continues.

If $\{\ell, m, C_a\}$ appears in the used anonymous certificate database or the Eq.(11) does not hold, the protocol terminates.

Task Assignment. After anonymous authentication, the MCS platform returns to the mobile user U_i a list of tasks near the location L and the corresponding costs for the user to choose. This information should be encrypted and decrypted with the secret key k agreed on in the authentication phase if the tasks are confidential. Without loss of generality, we consider one task assignment to the mobile user U_i . But it is easy to extend it to multiple tasks assignment. There are two cases when the MCS platform assigns a task to the mobile user U_i .

- If there is no task near the user, or the user is not interested in taking any task in the list, the MCS platform computes

$$B'_a = A'_a{}^{(2\ell+1)^{-1}d_a} \pmod{N} = r'^{(2\ell-1)} \cdot H(MCS, m')^{(2\ell+1)^{-1}d_a} \quad (12)$$

and returns B'_a to the mobile user U_i .

The user removes blindness by computing

$$C'_a = r'^{-(2\ell-1)} B'_a \pmod{N} = H(MCS, m')^{(2\ell+1)^{-1}d_a} \pmod{N} \quad (13)$$

In the end, the user obtains a new anonymous certificate $\{\ell, m', C'_a\}$, by which the user can still participate in MCS for times. The protocol terminates.

- Assume that the user has an anonymous reputation at the level λ , $\{\lambda, M, C_r\}$,

where $\lambda \geq 1$, and wishes to get an assignment to perform a task T , from the

list of tasks near his location L , the user randomly chooses integers R', M' from Z_N^* and computes

$$A'_r = R'^{(2\lambda-1)(2\lambda+1)(2\lambda+3)e_r)} H(MCS, M') \quad (14)$$

Then the user sends to the MCS platform a task request including

$$\{MCS, D, E_k(MCS, (\lambda, M, C_r), A'_r, T)\}.$$

Based on D in the task request, the MCS platform uses the corresponding secret key k to decrypt the request to obtain $(\lambda, M, C_r), A'_r, T$. Then the platform checks if the anonymous reputation $\{\lambda, M, C_r\}$ appears in the used anonymous reputation database and if

$$C_r^{(2\lambda+1)e_r} = H(MCS, M) \pmod{N}. \quad (15)$$

(i) If $\{\lambda, M, C_r\}$ does not appear in the used anonymous reputation database, the Eq.(15) holds, and the level of the reputation of any other user requesting the same task T is lower than λ , the platform assigns the task T to the mobile user U_i as follows. The platform computes

$$B'_a = A'_a{}^{(2\ell-1)^{-1}d_a} \pmod{N} = r'^{(2\ell+1)} \cdot H(MCS, m')^{(2\ell-1)^{-1}d_a} \quad (16)$$

and generates a signature δ on $\{MCS, (\ell, m, C_a), A'_a, (\lambda, M, C_r), A'_r, T\}$, that is,

$$\delta = H(MCS, (\ell, m, C_a), A'_a, (\lambda, M, C_r), A'_r, T)^{d_r} \quad (17)$$

Then the platform returns $\{(\ell - 1, B'_a), \delta\}$ to the user U_i and appends $(\lambda, M, C_r), A'_r$ to $\{D, k, (\ell, m, C), A', L\}_{aa}$ for next phase of communication with the user U_i .

The user removes blindness by computing

$$C'_a = r'^{-(2\ell+1)} B'_a \pmod{N} = H(MCS, m')^{(2\ell-1)^{-1} d_a} \pmod{N} \quad (18)$$

The user then obtains a new anonymous certificate $\{\ell - 1, m', C'_a\}$, by which the user can only participate in MCS for $\ell - 1$ times.

In addition, the user checks if

$$\delta^{e_r} = H(MCS, (\ell, m, C_a), A'_a, (\lambda, M, C_r), A'_r, T) \quad (19)$$

If the Eq.(19) holds, the task assignment for T is confirmed and the user then performs the task T . The protocol continues.

(ii) If $\{\lambda, M, C_r\}$ does not appear in the used anonymous reputation database and the Eq.(15) holds, but the level of the reputation of some user requesting the task T is higher than λ , the platform computes B'_a as the Eq.(12) and

$$B'_r = A_r'^{(2\lambda+1)^{-1} d_r} \pmod{N} = R'^{(2\lambda-1)(2\lambda+3)} \cdot H(MCS, M')^{(2\lambda+1)^{-1} d_r} \quad (20)$$

Then the platform informs the user that no task is available and returns (ℓ, B'_a) and (λ, B'_r) . With (ℓ, B'_a) , the user can compute C'_a as the Eq.(13), and with (λ, B'_r) , the user can compute

$$C'_r = R'^{-(2\lambda-1)(2\lambda+3)} B'_r \pmod{N} = H(MCS, M')^{(2\lambda+1)^{-1} d_r} \pmod{N} \quad (21)$$

In the end, the user obtains a new anonymous certificate $\{\ell, m', C'_a\}$, by which the user can still participate in MCS for times, and a new anonymous reputation still at the level λ , $\{\lambda, M', C'_r\}$. The protocol terminates. (iii) If $\{\lambda, M, C_r\}$ appears in the used anonymous reputation database or the Eq.(15) does not hold, the protocol terminates.

Report and Reward. After the mobile user U_i performs the task T , he writes a task report (e.g., the photos taken at an accident scene near the location L), encrypts it with the secret key k (agreed on in the authentication phase), and submits the encrypted report to the platform.

After receiving the encrypted report, the platform decrypts it with the secret key k to obtain the task report.

The MCS platform may assign the task T to multiple mobile users and compare the quality of their task reports. Based on the report from the mobile user U_i , the MCS platform rewards the mobile user with bitcoins. In this way, the MCS platform cannot even reveal the identity of the user when the user spends the bitcoin later. Next, the MCS platform generates a new anonymous reputation for the mobile user in three cases as follows:

- If the MCS platform thinks that the reputation of the mobile user should beupgraded, the MCS platform computes

$$B'_r = A'_r{}^{(2\lambda+3)^{-1}d_r} \pmod{N} = R'^{(2\lambda-1)(2\lambda+1)} H(MCS, M')^{(2\lambda+3)^{-1}d_r} \quad (22)$$

and returns $(\lambda + 1, B'_r)$ to the mobile user. The mobile user removes the blindness by computing

$$C'_r = R'^{-(2\lambda-1)(2\lambda+1)} B'_r \pmod{N} = H(MCS, M')^{(2\lambda+3)^{-1}d_r} \quad (23)$$

and obtains a new anonymous reputation at level $\lambda + 1$, $\{\lambda + 1, M', C'_r\}$. The protocol terminates.

- If the MCS platform thinks that the reputation of the mobile user should be kept without change, the MCS platform computes B'_r as the Eq.(20) and returns (λ, B'_r) to the users. Then the user removes the blindness by computing C'_r as the Eq.(21). In the end, the user obtains a new anonymous reputation still at level λ , $\{\lambda, M', C'_r\}$. The protocol terminates.
- If the MCS platform thinks that the reputation of the mobile user should be downgraded, the MCS platform computes

$$B'_r = A'_r{}^{(2\lambda-1)^{-1}d_r} \pmod{N} = R'^{(2\lambda+1)(2\lambda+3)} H(MCS, M')^{(2\lambda-1)^{-1}d_r} \quad (24)$$

and returns $(\lambda - 1, B'_r)$ to the mobile user. The mobile user removes the blindness by computing

$$C'_r = R'^{-(2\lambda+1)(2\lambda+3)} B'_r \pmod{N} = H(MCS, M')^{(2\lambda-1)^{-1}d_r} \quad (25)$$

and obtains a new anonymous reputation at level $\lambda - 1$, $\{\lambda - 1, M', C'_r\}$. The protocol terminates.

Link. After a mobile user has been authenticated anonymously by the MCS platform, if the mobile user does not care about linking his previous location with his current location for the purpose of efficiency, in particular when the user has not been assigned any task yet, the mobile user may choose to link his current location with his previous location with the secret key k agreed with the MCS platform in the authentication phase.

To do so, the mobile user computes

$$h = H(MCS, D, (\ell, m, C_a), A'_a, (\lambda, M, C_r), A'_r, t, k) \quad (26)$$

where t is a time stamp, and submits a re-authentication request $\{MCS, D, t, h\}$ to the MCS platform, through nearby base station for mobile communication. The base station forwards the request together with the location L to the MCS platform.

After receiving the link request, the MCS platform searches for D in its records and retrieves the corresponding secret key k , and then checks if the Eq.(26) holds. If so, the mobile user is re-authenticated. Note that the MCS platform keeps $\{D, k, (\ell, m, C_a), A'_a, L\}$ in the authentication phase.

Security and Performance Analysis

Due to the page limit, we provide security analysis in Appendix. In this section, we analyze the computation and communication overhead for our solution.

Performance of Our Scheme

The proposed LPP protocol consists of two main components that involve expensive operations on the participating parties: (1) Registration and (2) Protocol Execution. Within the second component, after a successful authentication of a mobile user U_i , two sub-components including (i) Task Assignment, and (ii) Report and Reward, may be executed, depending on whether or not U_i is assigned a task. In what follows, we focus on analyzing those expensive operations and the bandwidth consumption in each component.

Registration. During the registration for a mobile user U_i , after determining the number of times n this user is willing to participate in the protocol, U_i creates a random mask by raising a random element r to the power of $(2n+1)e_a$ modulo N , which is then multiplied by $H(MCS, m)$ to construct a blinded message A_a that will be forwarded as part of the registration request to the MCS platform P . Another blinded message A_r is created using a random element R and then sent to P likewise. After receiving the request, P creates two blind signatures B_a and B_r via raising A_a and A_r to the power of $(2n + 1)^{-1}d_a$ and $3^{-1}d_r$ modulo N , respectively. Knowing r and R , the mobile user U_i removes those previously attached random masks to create two signatures from the MCS platform. Two additional exponentiations modulo N are also needed to make sure the signatures are valid. In total, the number of exponentiations in Z_N^* performed by the MCS platform is 2, whereas it is 4 for U_i . On the other hand, at most $(4|Z_N| + |n|)$ bits are transmitted, where $|Z_N|$ and $|n|$ denote the bit-lengths of an element in Z_N^* and n , respectively.

Protocol Execution. A mobile user U_i participates in the LPP protocol by first authenticating himself with the MCS platform. Depending on the results of authentication and the ensuing task assignment, various computational and communicational costs will be incurred. To provide an upper bound on the overhead resulting from our protocol execution, in the following, we consider the case when U_i : (i) successfully passes the authentication; (ii) receives and accomplishes a task assignment from the MCS platform, and then (iii) updates his anonymous certificate and reputation for the next participation.

During the first stage, U_i encrypts a valid signature on (MCS, m) received from the MCS platform using a key k derived from raising a random element s in Z_N^* to the power of e_a . A blinded message A'_a for the next authentication is prepared by U_i as well, which costs 1 exponentiation. Both the encrypted signature and the blinded message are sent to the MCS platform, which in turn computes the decryption key k via raising the concealed key to its d_a -th power to retrieve the anonymous certificate C_a . The MCS platform then verifies the validity of C_a by raising it to the power of $(2^\ell + 1)e_a$. To sum up, both the

MCS platform and U_i carry out 2 exponentiations in Z_N^* , and at most $(4|Z_N| + |L| + |\ell| + 2|MCS|)$ bits are transmitted, where, and $|MCS|$ denote the

respective bit-lengths of L , ℓ , and MCS .

Once the verification succeeds, U_i sends to the MCS platform a task request, which consists of U_i 's reputation λ along with its signature C_r encrypted by the same key k in the previous stage. A blinded message A_r' that incurs 1 exponentiation on U_i for updating the reputation in the next task assignment is sent with C_r as well. After obtaining λ and C_r using k , the MCS platform verifies the validity of the signature, which takes 1 exponentiation. When C_r is valid and the reputation submitted by U_i is the highest one, the MCS platform generates and sends to U_i the signature σ on the assigned task T as well as the blinded signature $B'_{aon}(MCS, m')$ for U_i 's next authentication, each costs 1 exponentiation, respectively. Other than checking the validity of σ , U_i also makes sure the anonymous certificate C'_a is valid after removing the corresponding random mask $r^{(2\ell+1)}$ from B'_a . Three additional exponentiations are thus performed by U_i . In total, the MCS platform performs 3 exponentiations and U_i performs $4|Z_N| + 2|MCS| + |\ell| + |\lambda| + |T|$ exponentiations. The number of bits exchanged is at most (6), where $|\lambda|$, and $|T|$ correspond to the bit-lengths of λ , and T .

After the assigned task T is finished, depending on the submitted task report from U_i , the MCS platform computes the blind signature B'_r on U_i 's updated reputation $\lambda' \in \{\lambda-1, \lambda, \lambda+1\}$, which takes 1 exponentiation. To create an updated anonymous reputation C'_r , U_i first produces $R^{(2\lambda-1)(2\lambda+1)}$, $R^{(2\lambda-1)(2\lambda+3)}$, or $R^{(2\lambda+1)(2\lambda+3)}$ modulo N to remove the random mask from C'_r , when the MCS platform decides to upgrade, maintain, or downgrade the reputation of U_i . The verification of the updated reputation is performed by U_i as usual, which costs 1 exponentiation. Overall, the MCS platform carries out 1 exponentiation and U_i performs 2 exponentiations. At most $(|Z_N| + |\lambda'| + |T_R|)$ bits are transmitted in this phase, where $|\lambda'|$ and $|T_R|$ represent the bit-lengths of the updated reputation λ and task report T_R , respectively.

Comparison

Let us theoretically compare the computation and communication complexities of the registration and authentication phases in our protocol with some existing protocols, e.g., Ramzan et al.'s protocol [26], and Blanton's protocol [3], both of which support anonymous authentication. The protocol in [26] allows each user access to the service for a fixed number of times, whereas a user in the latter protocol [3] is able to obtain the service before the specified expiration time. Since our protocol also incurs costs stemming from reputation updates and task assignments that do not exist in the other two schemes, for the sake of comparison, we exclude those costs when comparing our protocol with the other two, i.e., we only consider operations directly related to registration and authentication in the following comparisons.

Moreover, both Ramzan et al.'s and Blanton's protocols invoke zero-knowledge protocols as building blocks and thus the related security parameters have to be specified when necessary. In what follows, we use κ_1 , κ_2 , κ_3 to denote the security

parameters associated with the soundness, completeness, and statistical zero-knowledge of a protocol, respectively. Alternatively, κ_4 is used to represent the bit-length of a user authentication key, when it is explicitly specified in a protocol. We also use n and n' to denote the maximum number of accesses allowed and the maximum expiration time in a system, respectively, and let $|n'|$ and $|n''|$ denote their corresponding bit-lengths.

To compare the computational complexities of different protocols, for each protocol, we count the expensive operations, e.g., exponentiations or pairing operations in the corresponding groups. We will use the name of a group to denote the exponentiations under this group. For instance, our scheme requires exponentiations in Z_N^* , which are also necessary to construct and verify range proofs [7] in Ramzan et al.'s protocol. Additionally, based on an RSA modulus N , a large prime $P' = jN + 1$ for some integer j has to be generated to enable the creation and verification of signatures of knowledge [6] in a cyclic multiplicative subgroup of $Z_{P'}^*$ for Ramzan et al.'s scheme. In Blanton's protocol, two cyclic groups G_1 and G_T of prime order Q equipped with an efficient bilinear pairing are needed instead to create blind signatures and the corresponding proofs of knowledge. A bilinear pairing is a map $e : G_1 \times G_1 \rightarrow G_T$ such that for all $g, h \in G_1$, $a, b \in \mathbb{Z}$, it holds that $e(g^a, h^b) = e(g, h)^{ab}$. The term Pp will be used to denote such an operation. Moreover, a term nG^x indicates n multiexponentiations in the group G with x bases.

On the other hand, when comparing the communication complexities, $|G|$ is used to signify the size of a group element in the group G . We will also use $|X|$ to denote the bit-length to represent a variable X when it is clear from the context. For instance, we use $|MCS|$ to denote the bit-length of MCS .

We first compare these protocols in terms of computational complexities in the registration stage. Table 1 lists the expensive operations performed in each protocol on the server and user sides, respectively².

Table 1. Computation comparison (registration)

Protocol	Server	User	Communication cost
Ours	$2Z_N$	$4Z_N$	$ n' + 4 Z_N $
Ramzan et al. [26]	$(\kappa_1 + 4)Z_N + (\kappa_1 + 1)Z_{P'}$	$(\kappa_1 + 7)Z_N + (\kappa_1 + 2)Z_{P'}$	$(\kappa_1 + 3) Z_N + 2 Z_{P'} + 4\kappa_1 + 2\kappa_2 + \kappa_3 + 2\kappa_4 + 1$
Blanton [3]	$9G_1 + G_1^2 + G_1^3$	$G_1 + 2G_{21} + G_T + G_{3T} + 20Pp$	$11 G_1 + 3 Z_Q + \kappa_1$

It can be seen that our scheme is much more efficient than Ramzan et al.'s because the number of exponentiations carried out in the zero-knowledge protocols they adopt depends on κ_1 , the soundness of the zero-knowledge protocols. To be exact, this security parameter guarantees that a cheating prover can only succeed in

² Note that we will use Server to denote our MCS platform when focusing our comparisons on registration and authentication.

constructing a valid zero-knowledge proof with probability at most $1/2^{\kappa_1}$. In practice, κ_1 should be at least 40 to thwart a cheating prover. The computational complexity of Blanton's protocol does not depend on κ_1 as described above even though they also adopt zero-knowledge proofs as building blocks. However, our scheme still requires much less computation in terms of the number of exponentiations. It shall be clear in Sect.5.3 that our scheme is much more efficient empirically.

As for the communication complexities, we provide the information in Table1 as well. Our scheme is the most efficient one among these three. Ramzan et al.'s protocol consumes the highest bandwidth resulting from the invocation of zero-knowledge protocols. Although Blanton's protocol does not incur such a high communication cost, it still requires exchanging at least 11 group elements in G_1 , each of which is of similar bit-length to an element in Z_N^* if we adopt the parameter setting described at the beginning of Sect.5.3.

In Table2, we provide a comparison of these three schemes based on the computational costs incurred in the authentication phase. Again, our protocol requires the lowest number of exponentiations during authentication on both the server and user sides. As seen previously, Ramzan's scheme incurs a much higher computational cost, which grows linear in κ_1 . The number of exponentiations performed in Blanton's protocol, on the other hand, depends on $|n''|$, the bitlength used to represent the maximum expiration time in a system. This is mainly because this scheme has to invoke a zero-knowledge protocol that proves a committed number lies within a range in a bitwise manner. We can see that even when $|n''|$ is set to a small integer, e.g., 10, this protocol still needs much more exponentiations than ours does.

Table 2. Computation comparison (authentication)

Protocol	Server	User	Communication cost
Ours	$3Z_N$	$4Z_N$	$4 Z_N + 2 MC_S + n' $
Ramzan et al. [26]	$\kappa_1 Z_N + (\kappa_1 + 11)Z_{P'}$	$\kappa_1 Z_N + (\kappa_1 + 13)Z_{P'}$	$2 Z_{P'} + (\kappa_1 + 5) Z_N + 3\kappa_1$
Blanton [3]	$8G_1 + 2G_1^3 + (2 n'' + 1)G_1^2 + G_T + G_T^2 + G_T^3 + 6Pp$	$((n'' + 11)G_1 + (n'' + 2)G_1^2 + G_T + 2G_T^2 + 2G_T^3 + 26Pp$	$(2 n'' + 21) G_1 + G_T + (4 n'' + 7) Z_Q + (n'' + 2)\kappa_1$

Lastly, we compare the communication costs in Table2. Our protocol consumes the lowest bandwidth because its bandwidth consumption is affected by neither κ_1 nor $|n''|$.

Experiments

To assess the efficiency of various schemes, we implement our and Ramzan et al.'s protocols using the GNU MP library³. The RSA modulus N needed in our and Ramzan et al.'s protocol is of 1,024-bit long, a product of two safe primes $p = 2p' + 1$ and $q = 2q' + 1$ of equal bit-length, where p and q are both primes. As for the prime P that will be additionally required in Ramzan et al.'s protocol, we set $P' = 2N + 1$ to estimate the lower bound on the computational complexity of exponentiations in $\mathbb{Z}_{P'}^*$. On the other hand, for Blanton's protocol, we use the Pairing-Based Cryptography Library (PBC) [22] to instantiate an A512 group of order Q , a 160-bit prime. In our experiments, we set those four security parameters $|n'|$ as $\kappa_1 = 80$, $\kappa_2 = \kappa_3 = 40$, and $\kappa_4 = 160$, when necessary⁴. The bit-length to represent the maximum number of accesses allowed in a system is set to 80 and the bit-length $|n''|$ to represent the maximum expiration time in Blanton's protocol is set to 10.

The machine we use has an Intel i7-4770HQ with 16GBytes of RAM and the operating system installed is an Ubuntu 14.04 LTS. All the source code for the prototype of server and user is implemented in C.

Table 3. Registration and authentication time comparison of user (in ms)

Protocol	Registration	Authentication
Ours	0.822	1.302
Ramzan et al. [26]	48.598	51.378
Blanton [3]	13.312	62.836

The computational cost incurred on the MCS platform and a mobile user in each stage that involves expensive operations, i.e., exponentiations in $\mathbb{Z}_N^*$, are 0.822ms and 1.64ms, respectively, in registration stage, and 2.466ms and 2.604ms, respectively, in protocol execution stage. We can see that our protocol incurs very little cost on both the server and user in each phase involving expensive operations. On average, the time spent in a single stage on either party is below 2.7ms. It will also be clearer that our scheme is much more efficient when it is empirically compared with the other two.

Let us now compare our protocol with the other two in terms of the computational overhead on the user in the registration and authentication phases. The detailed information is given in Table 3, according to which we can see that our scheme requires a minimum amount of computation on the user in both phases. Specifically, each invocation of our registration or authentication protocol takes less than 1.4ms to finish on average. On the other hand, as analyzed in Sect. 5.2, Ramzan et al.'s protocol incurs the highest overhead during registration. We also find out that even when the maximum expiration time is as small as $2^{|n''|} = 2^{10}$, a

³ <https://gmplib.org/>.

⁴ We adopt the suggested setting of κ_1 , κ_2 , and κ_3 in [4] and [6].

user in Blanton's scheme still has to spend at least 62.8ms to complete the authentication, which is at least 44 times higher than ours.

We have also conducted experiments to measure the computational costs imposed on the server by varying the number of users it needs to process during registration and authentication. The results are given in Figs.1 and 2, respectively. In all of these protocols, the computational cost grows linearly in the number of users served. However, Ramzan et al.'s and Blanton's schemes grow much faster than ours due to their dependence on either the soundness security parameter κ_1 or the bit-length of maximum expiration time $|n''|$. To be precise, when there are 10,000 users waiting to be authenticated, the other two protocols require at least 8min to accomplish the task, whereas it only takes our protocol less than 13s, which clearly indicates that our scheme is much more scalable.

Fig.1. Registration time comparison **Fig.2.** Authentication time comparison of server when varying number of users son of server when varying number of users

Conclusions

In this paper, we have described a location privacy-preserving (LPP) protocol for MCS. The proposed protocol overcomes the trusted third party, linkability and key sharing problems of the existing solutions for location privacy protection in MCS. In addition, we introduce a new concept of anonymous reputation. The MCS platform can refer to the anonymous reputation of a mobile user when assigning tasks to the user. Based on the report submitted by the user, the MCS platform can upgrade or downgrade the reputation of the user.

Acknowledgements. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative, and Australian Research Council (ARC) Discovery Projects DP160100913 & DP180103251.

Appendix: Security Analysis

The proposed LPP protocol is based on the blind signature scheme. According to [17], a blind digital signature scheme is secure if for all probabilistic polynomialtime (PPT) algorithms A , the following two considerations hold.

Blindness Property: Let b is a random bit which is kept secret from A. A executes the following experiment (where A controls the signer, but not the user, and tries to predict b):

Step 1: $(pk, sk) \leftarrow \text{Gen}(1^k)$

Step 2: $(m_0, m_1) \leftarrow A(1^k, pk, sk)$ (i.e. A produces two documents, where (m_0, m_1) are by convention lexicographically ordered and may even depend on pk and sk).

Step 3: We denote by (m_b, m_{1-b}) the same two documents (m_0, m_1) , ordered according to the value of bit b , where the value of b is hidden from A. $A(1^k, pk, sk, m_0, m_1)$ engages in two parallel (and arbitrarily interleaved) interactive protocols, the first with $\text{User}(pk, m_b)$ and the second with

$\text{User}(pk, m_{1-b})$.

Step 4: If the first user outputs on his private tape $\sigma(m_b)$ (i.e., does not output fail) and the second user outputs on his private tape $\sigma(m_{1-b})$ (i.e., also does not output fail) then A is given as an additional input $(\sigma(m_b), \sigma(m_{1-b}))$ ordered according to the corresponding (m_0, m_1) order. (We remark that we do not insist that this happens, and either one or both users may output fail).

Step 5: A outputs a bit b (given his view of steps 1 through 3, and if conditions are satisfied, of step 4 as well).

Then the probability, taken over the choice of b , over coin-flips of keygeneration algorithm, the coin-flips of A, and (private) coin-flips of both users (from step 3), $b' = b$ is negligibly close to $1/2$.

Unforgeability Property: A executes the following experiment (where A controls the user, but not the signer, and tries to get one more signature):

Step 1: $(pk, sk) \leftarrow \text{Gen}(1^k)$

Step 2: $A(pk)$ engages in polynomially many (in k) adaptive, parallel and arbitrarily interleaved interactive protocols with polynomially many copies of $\text{Signer}(pk, sk)$, where A decides in an adaptive fashion when to stop. Let j denote the number of executions, where the signer outputted completed in the end of Step 2.

Step 3: A outputs a collection $\{(m_1, \sigma(m_1)), (m_2, \sigma(m_2)), \dots, (m_j, \sigma(m_j))\}$ subject to the constraint the all $(m_i, \sigma(m_i))$ for $1 \leq i \leq j$ are all accepted by

$\text{Verify}(pk, m_i, \sigma(m_i))$.

Then the probability, taken over coin-flips of key - generation algorithm, the coin flips of A , and over the (private) coin-flips of the *Signer*, that $j >$ is negligible.

For the following security analysis, we make an assumption, which can reasonably be expected to hold in practice. We assume that on average the users have the same total access times (i.e., during the registration, n is the same for every user), and access the MCS system with the same frequency. This implies that at every given point in time, there will be a similar number of users that have each possible remaining access times (i.e.,). In other words, the number of remaining access times for a user is equally likely to be any number between 1 and n (i.e., $1 \leq \ell \leq n$). In addition, we assume the Chaum's blind signature scheme [8] is secure in terms of blindness and forgeability.

During MCS, the platform learns one thing. He sees the anonymous certificates and anonymous reputations, i.e., the blind signatures, used in MCS. We claim that the MCS platform learns nothing from the blind signatures themselves, and only the number of the participation of the mobile user and the reputation level of the mobile user in MCS.

At first, let us analyse the anonymity of the proposed protocol with a game according to Definition 1 in Sect. 2. For this security analysis, we assume that the MCS platform is malicious and tries to identify the mobile user.

Given two mobile users U_0 and U_1 , assume that the MCS platform runs the registration protocol with them, respectively, to issue blind signatures to them for anonymous authentication.

Let us choose a bit b randomly.

In the authentication phase, the mobile user U_b submits the authentication request $\{MCS, D_b, E_{k_b}(MCS, (\ell_b, m_b, C_{a,b}), A'_{a,b})\}$ to the platform. The platform can derive the secret key k_b from D_b with its private key d_a and perform decryption to obtain the anonymous certificate $\{\ell_b, m_b, C_{a,b}\}$. Due to the blindness property of the Chaum's blind signature, the platform cannot tell if the blind signature is from the mobile user U_0 or U_1 .

In the task assignment phase, the mobile user U_b submits to the MCS platform a task request $\{MCS, D_b, E_{k_b}(MCS, (\lambda_b, M_b, C_{r,b}), A'_{r,b}, T_b)\}$. With k_b corresponding to D_b , the platform performs decryption to obtain the anonymous reputation $\{\lambda_b, M_b, C_{r,b}\}$. Due to the blindness property of the Chaum's blind signature, the MCS platform cannot tell if the blind signature is from the mobile user U_0 or U_1 .

In the report and reward phase, the mobile user does not submit any blind signature to the MCS platform. The MCS platform has no way to distinguish the mobile users in this phase.

Based on the above security analysis, according to Definition 1 for anonymity, we conclude that

Theorem 1. The proposed LPP protocol has anonymity if the Chaum's blind signature has blindness.

Next, let us analyse the unlinkability of the proposed protocol with a game.

Given two mobile users U_0 and U_1 , assume that the platform runs the protocol with U_0 and U_1 , respectively, and keeps two anonymous certificates and two anonymous reputations: $\{\ell_0, m_0, C_{a,0}\}$ and $\{\ell_0, M_0, C_{a,0}\}$ from U_0 , $\{\ell_1, m_1, C_{a,1}\}$ and $\{\ell_1, M_1, C_{a,1}\}$ from U_1 .

Next, let us choose a bit b randomly. User U_b runs the protocol with the MCS platform again and provides the MCS platform with anonymous certificate and anonymous reputation: $\{\ell'_b, m'_b, C'_{a,b}\}$ and $\{\ell'_b, M'_b, C'_{a,b}\}$.

Due to the blindness property of the Chaum's blind signature, the MCS platform cannot tell if the blind signatures $\{\ell'_b, m'_b, C'_{a,b}\}$ and $\{\ell'_b, M'_b, C'_{a,b}\}$ are from the mobile user U_0 or U_1 . Based on the above analysis, according to Definition 2 for unlinkability, we conclude that

Theorem 2. The proposed LPP protocol has unlinkability if the Chaum's blind signature has blindness.

At last, let us analyse the unforgeability of the proposed protocol with a game. For this analysis, we assume a group of mobile users are malicious. For simplicity, we consider anonymous certificates only at first and then we can easily extend the security analysis for anonymous reputation, because both of them are blind signatures anyway.

In the proposed LPP protocol, a valid anonymous certificate takes the form of $\{\ell, m, C = H(MCS, m)^{(2^\ell+1)^{-1}d_a}\}$ for $\ell = 1, 2, \dots$. Assume that the adversary is given t valid anonymous certificates $\{\ell_i, m_i, C_i\}$ for $i = 1, 2, \dots, t$, if the

adversary can generate a new anonymous certificate, which is different from the given t anonymous certificates, he wins the game.

In the given t valid anonymous certificates, if $\ell_1 = \ell_2 = \dots = \ell_t = \ell$, the adversary cannot forge any more new anonymous certificate because the Chaum's blind signature for the public key $(2^\ell + 1)e_a$ has unforgeability.

In the given t valid anonymous certificates, if we group certificates on the basis of the public key $(2^\ell + 1)e_a$, the adversary cannot forge any more new certificate in any group with the same public key because the Chaum's blind signature for the public key $(2^\ell + 1)e_a$ has unforgeability.

Now let us consider the possibility of forging a new anonymous certificate across the groups, i.e., how to forge a new anonymous certificate $\{\ell', m', C' = H(MCS, m')^{(2^{\ell'}+1)^{-1}d_a}\}$ with two anonymous certificates $\{\ell_1, m_1, C_1\}$ such that $C_1 = H(MCS, m_1)^{(2^{\ell_1}+1)^{-1}d_a} \pmod{N}$ and $\{\ell_2, m_2, C_2\}$ such that $C_2 =$

$$H(MCS, m_2)^{(2^{\ell_2}+1)^{-1}d_a} \pmod{N}, \text{ where } \ell_1 \neq \ell_2.$$

Because the hash function H is collision-resistant, from $H(MCS, m_1)^{(2^{\ell_1}+1)^{-1}d}$ and $H(MCS, m_2)^{(2^{\ell_2}+1)^{-1}d}$, it is hard to forge a new anonymous certificate (ℓ', m', C') as follows.

$$\begin{aligned}
& - C' = H(MCS, m_1)^{(2\ell'+1)^{-1}d_a \pmod N} \text{ for some } \ell', \text{ such that } \ell' \neq \ell_1. \\
& - C' = H(MCS, m_2)^{(2\ell'+1)^{-1}d_a \pmod N} \text{ for some } \ell', \text{ such that } \ell' \neq \ell_2. \\
& - C' = H(MCS, m')^{(2\ell'+1)^{-1}d_a \pmod N} \text{ for some } \ell', \text{ such that } m' \neq m_1 \text{ and } m' \neq m_2.
\end{aligned}$$

In view of this, we conclude that

Theorem 3. The proposed LPP protocol has unforgeability if the Chaum's blind signature has unforgeability and the hash function H is collision-resistant.

References

1. Abe, M., Fujisaki, E.: How to date blind signatures. In: Kim, K., Matsumoto, T. (eds.) ASIACRYPT 1996. LNCS, vol. 1163, pp. 244–251. Springer, Heidelberg (1996). <https://doi.org/10.1007/BFb0034851>
2. Bellavista, P., Corradi, A., Foschini, L., Ianniello, R.: Scalable and cost-effective assignment of mobile crowdsensing tasks based on profiling trends and prediction: the participact living lab experience. *Sensors* **15**(8), 18613–18640 (2015)
3. Blanton, M.: Online subscriptions with anonymous access. In: Proceedings of ASIACCS 2008, pp. 217–227 (2008)
4. Boudot, F.: Efficient proofs that a committed number lies in an interval. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 431–444. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45539-6_31
5. Brickell, E., Camenisch, J., Chen, L.: Direct anonymous attestation. In: Proceedings of 11th ACM Conference on Computer and Communication Security, pp. 132–145 (2004)
6. Camenisch, J., Stadler, M.: Efficient group signature schemes for large groups. In: Kaliski, B.S. (ed.) CRYPTO 1997. LNCS, vol. 1294, pp. 410–424. Springer, Heidelberg (1997). <https://doi.org/10.1007/BFb0052252>
7. Chan, A., Frankel, Y., Tsiounis, Y.: Easy come — easy go divisible cash. In: Nyberg, K. (ed.) EUROCRYPT 1998. LNCS, vol. 1403, pp. 561–575. Springer, Heidelberg (1998). <https://doi.org/10.1007/BFb0054154>
8. Chaum, D.: Blind signatures for untraceable payments. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) Advances in Cryptology, pp. 199–203. Springer, Boston, MA (1983). https://doi.org/10.1007/978-1-4757-0602-4_18
9. Chaum, D., van Heyst, E.: Group signatures. In: Davies, D.W. (ed.) EUROCRYPT 1991. LNCS, vol. 547, pp. 257–265. Springer, Heidelberg (1991). https://doi.org/10.1007/3-540-46416-6_22
10. Christin, D.: Privacy in mobile participatory sensing: current trends and future challenges. *J. Syst. Softw.* **116**, 57–68 (2016)
11. Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., Yu, T.: Differentially private spatial decompositions. In: Proceedings of ICDE 2012, pp. 20–31 (2012)
12. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer, Berlin (2002). <https://doi.org/10.1007/978-3-662-04722-4>
13. Dwork, C.: Differential privacy. In: Bugliesi, M., Preneel, B., Sassone, V., Wegener, I. (eds.) ICALP 2006. LNCS, vol. 4052, pp. 1–12. Springer, Heidelberg (2006). https://doi.org/10.1007/11787006_1
14. Ganti, R.K., Ye, F., Lei, H.: Mobile crowdsensing: current state and future challenges. *IEEE Commun. Mag.* **49**(11), 32–39 (2011)

15. Guo, B., Calabrese, F., Miluzzo, E., Musolesi, M.: Mobile crowd sensing: part 1. *IEEE Commun. Mag.* **52**(8), 20–21 (2014)
16. Guo, B., Calabrese, F., Miluzzo, E., Musolesi, M.: Mobile crowd sensing: part 2. *IEEE Commun. Mag.* **52**(10), 76–77 (2014)
17. Juels, A., Luby, M., Ostrovsky, R.: Security of blind digital signatures. In: Kaliski, B.S. (ed.) *CRYPTO 1997*. LNCS, vol. 1294, pp. 150–164. Springer, Heidelberg (1997). <https://doi.org/10.1007/BFb0052233>
18. Kantarci, B., Glasser, P.M., Foschini, L.: Crowdsensing with social network-aided collaborative trust scores. In: *Proceedings of IEEE Global Communication Conference (GLOBECOM)*, pp. 1–6 (2015)
19. Kantarci, B., Carr, K.G., Pearsall, C.D.: SONATA: social network assisted trustworthiness assurance in smart city crowdsensing. *Int. J. Distrib. Syst. Technol.* **7**(1), 59–78 (2016)
20. Kapadia, A., Triandopoulos, N., Cornelius, C., Peebles, D., Kotz, D.: AnonySense: opportunistic and privacy-preserving context collection. In: *Proceedings of 6th International Conference on Mobile System, Applications and Services (MobiSys)*, pp. 280–297 (2008)
21. Konidala, D.M., Deng, R.H., Li, Y., Lau, H.C., Fienberg, S.E.: Anonymous authentication of visitors for mobile crowd sensing at amusement parks. In: Deng, R.H., Feng, T. (eds.) *ISPEC 2013*. LNCS, vol. 7863, pp. 174–188. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38033-4_13
22. Lynn, B.: On the implementation of pairing-based cryptosystems. Stanford University (2007)
23. Navas, J.C., Imielinski, T.: GeoCast - geographic addressing and routing. In: *Proceedings of ACM International Conference on Mobile Computing and Networking*, pp. 66–76 (1997)
24. Pouryazdan, M., Kantarci, B., Soyata, T., Song, H.: Anchor-assisted and votebased trustworthiness assurance in smart city crowdsensing. *IEEE Access* **4**, 529–541 (2016)
25. Pouryazdan, M., Kantarci, B., Soyata, T., Foschini, L., Song, H.: Quantifying user reputation scores, data trustworthiness, and user incentives in mobile crowdsensing. *IEEE Access* **5**, 1382–1397 (2017)
26. Ramzan, Z., Ruhl, M.: Protocols for anonymous subscription services (2000). (Unpublished Manuscript)
27. Ren, J., Zhang, Y., Zhang, K., Shen, X.S.: SACRM: social aware crowdsourcing with reputation management in mobile sensing. *Comput. Commun.* **65**, 55–65 (2015)
28. Shina, M., Cornelius, C., Peebles, D., Kapadia, A., Kotz, D., Triandopoulos, N.: AnonySense: a system for anonymous opportunistic sensing. *Pervasive Mobile Comput.* **7**, 16–30 (2011)
29. Sweeney, L.: k-anonymity: a model for protecting privacy. *Int. J. Uncertainty Fuzziness Knowl.-Based Syst.* **10**, 557–570 (2002)
30. To, H., Ghinita, G., Shahabi, C.: A framework for protecting worker location privacy in spatial crowdsourcing. In: *Proceedings of VLDB 2014*, pp. 919–930 (2014)
31. Vergara-Laurens, I.J., Jaimes, L.G., Labrador, M.A.: Privacy-preserving mechanisms for crowdsensing: survey and research challenges. *IEEE IoT J.* **4**(4), 855–869 (2017)
32. Wang, X., Liu, Z., Tian, X., Gan, X., Guan, Y., Wang, X.: Incentivizing crowdsensing with location-privacy preserving. *IEEE Trans. Wirel. Commun.* **16**(10), 6940–6952 (2017)