Journal of Communications and Information Networks, Vol.2, No.2, Jun. 2017
DOI: 10.1007/s41650-017-0017-7

Research paper

Special Issue on Internet of Vehicle

Privacy-friendly spatial crowdsourcing in vehicular networks

Cheng Huang[1]*, Rongxing Lu[2], Hui Zhu[3]

1. Department of Electrical and Computer Engineering, University of Waterloo, Waterloo ON N2L 3G1, Canada
2. Faculty of Computer Science, University of New Brunswick, Fredericton NB E3B 5A3, Canada
3. School of Cyber Engineering, Xidian University, Xi'an 710126, China

* Corresponding author, Email: c225huan@uwaterloo.ca

Abstract: With the evolution of conventional VANETs (Vehicle Ad-hoc Networks) into the IoV (Internet of Vehicles), vehicle-based spatial crowdsourcing has become a potential solution for crowdsourcing applications. In vehicular networks, a spatial-temporal task/question can be outsourced (i.e., task/question relating to a particular location and in a specific time period) to some suitable smart vehicles (also known as workers) and then these workers can help solve the task/question. However, an inevitable barrier to the widespread deployment of spatial crowdsourcing applications in vehicular networks is the concern of privacy. Hence, We propose a novel privacy-friendly spatial crowdsourcing scheme. Unlike the existing schemes, the proposed scheme considers the privacy issue from a new perspective according that the spatial-temporal tasks can be linked and analyzed to break the location privacy of workers. Specifically, to address the challenge, three privacy requirements (i.e. anonymity, untraceability, and unlinkability) are defined and the proposed scheme combines an efficient anonymous technique with a new composite privacy metric to protect against attackers. Detailed privacy analyses show that the proposed scheme is privacy-friendly. In addition, performance evaluations via extensive simulations are also conducted, and the results demonstrate the efficiency and effectiveness of the proposed scheme.

Keywords: spatial crowdsourcing, vehicular networks, location privacy, IoV (Internet of Vehicles), privacy metric

Citation: C. Huang, R. X. Lu, H. Zhu. Privacy-friendly spatial crowdsourcing in vehicular networks [J]. Journal of communications and information networks, 2017, 2(2): 59-74.

1 Introduction

Spatial crowdsourcing, as a new paradigm, has been proposed recently to help solve spatial-temporal questions by the public. In other words, the goal of spatial crowdsourcing is to outsource a set of particular tasks (i.e., tasks related to a specific area/region/road in a given period) to a group of suitable workers who will perform these tasks during that time. Generally, these crowdsourcing tasks could be categorized into two types: sensing-based and human-intelligence-based tasks. For sensing-based tasks, the workers can easily work as moving sensors to conduct tasks, such as air quality sensing in the city, clicking photos and recording audios/videos at some specific places. Comparatively, human-intelligence-based tasks are more complex because each worker is required to not only be equipped with multiple sensors but also use his/her intelligence to make decisions or give suggestions,

Manuscript received Mar. 09, 2017; accepted Apr. 10, 2017
This work is supported by the National Natural Science Foundation of China (No. 6167241), the National Basic Research Plan in Shannxi Province of China (2016JM6007).

such as recommending nearby delicious restaurants and reporting nearby traffic conditions. In real-world scenarios, there exist many spatial crowdsourcing applications. The typical applications include MediaQ (video clips)[1], Foursquare (local hotspots) (https://foursquare.com.), and Waze (traffic monitor) (https://www.waze.com.). In general, spatial crowdsourcing applications have drawn increasing attention from industry.

In spatial crowdsourcing applications, mobile device users, such as smartphone users are regarded as the most common workers. Although smartphone-based spatial crowdsourcing has proven its potential in dealing with complex spatial-temporal issues, it is undeniable that the vehicle-based spatial crowdsourcing is evolving rapidly with the advent of ITS (intelligent transport system)[2]. The evolution of conventional VANET into the IoV indicates that the vehicular networks may support a more general vehicle to X (X can be vehicles, roadside units, sensors, human, or Internet) communication by integrating the inter-vehicle network (i.e., vehicles' interconnection), the intra-vehicle network (also named connected vehicles), and the vehicular mobile Internet (each vehicle is seen as a wheeled mobile node). However, the modern smart vehicle is able to be equipped with a large number of on-board sensors and a powerful processor, which can help obtain information about the surroundings (e.g., temperature, humidity, and pollution gases). Moreover, a smart vehicle's mobility characteristics appropriately match the spatial-temporal requirements for crowdsourcing tasks. Hence, vehicle networks could be a proper infrastructure in the urban area for spatial crowdsourcing applications (Fig. 1).

A vehicle-based spatial crowdsourcing system usually consists of three major roles: workers, crowdsourcing platform, and users. The users upload the spatial-temporal tasks to the platform, where the workers are a group of smart vehicles that contribute to the uploaded crowdsourcing tasks. However, there exist some underlying security and privacy concerns, and these concerns seriously inhibit the widespread deployment of spatial crowdsourcing applications in vehicular networks. Concretely, as a spatial-temporal task requires the workers to upload their locations in real time, the platform can easily infer the workers' locations, which raises serious location privacy concerns of the workers.

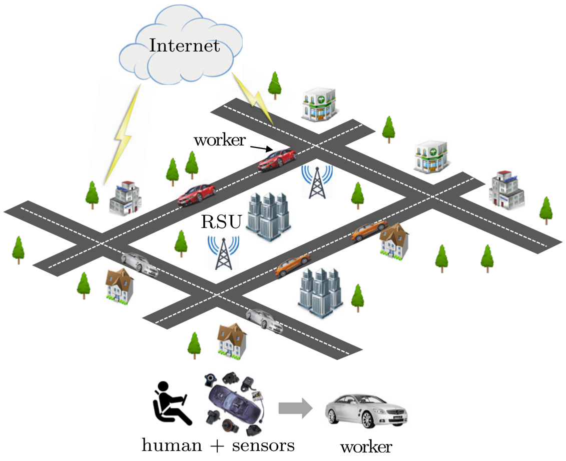


Figure 1 Spatial crowdsourcing in vehicular networks

In recent years, several related schemes and approaches have been proposed to cope with the location privacy issue for spatial crowdsourcing[3-7]. Most of these schemes address the location privacy issue by masking the location information based on differential privacy approach (e.g., by adding noise to the uploaded location data). However, a critical fact is ignored and never discussed in these schemes. Even if the uploaded location is masked, spatial-temporal tasks performed by the workers can also be linked and analyzed to compromise the workers' location privacy, especially considering that the vehicles are workers on the fixed road network. In other words, regardless of the type of techniques used by the workers to mask their locations, the platform knows that these workers will be physically present at some certain places to complete the task in a short period. For instance, a vehicle plans to accept crowdsourcing tasks on its way from office to home, and chooses several jobs in the city, as shown in Fig. 2. The spatial-temporal tasks could be easily linked and analyzed by the platform to recover the trajectories of this vehicle according to the tasks accepted by the vehicle. There are pieces of evidence that are likely to reconstruct the path even with sparse location data of vehicles[8]. Moreover, the reconstructed trajectories can be utilized to identify a particular smart vehicle even if it uses pseudonyms to hide its real identity.

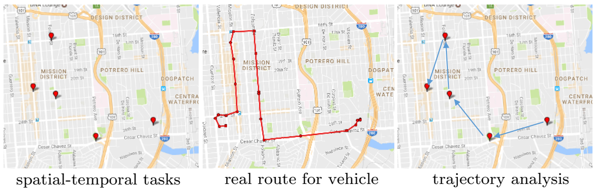


Figure 2 Linking of spatial-temporal tasks

To address the above-mentioned privacy challenge for spatial crowdsourcing system in vehicular networks, in this paper, we propose a novel privacy-friendly spatial crowdsourcing scheme. In our system, each vehicle can manage their location privacy while performing the crowdsourcing tasks as an anonymous mobile worker with minimal privacy disclosure and finally gain the corresponding rewards. We regard the contributions of this paper to be threefold:

- We propose a composite privacy metric for evaluating privacy exposure for spatial crowdsourcing applications in vehicular networks; this involves three parts: the frequency of changing pseudonym k, degree of task distribution $degree_t$, and degree of task similarity $degree_s$. Based on the proposed privacy metric, the workers can select suitable tasks and control their privacy leakage through different settings. We also analyze the real taxi-trace dataset[9] to demonstrate the effectiveness of the proposed privacy metric.

- We propose a privacy-friendly spatial crowdsourcing scheme to achieve location privacy by combining an efficient anonymous technique with the proposed privacy metric; this allows the smart vehicles, that is, workers to anonymously accept the crowdsourcing job, report the results, and obtain the corresponding rewards. Moreover, the proposed scheme protects workers' location privacy in terms of not only anonymity but also untraceability and unlinkability.

- To demonstrate the utility of our proposed scheme, we implemented the proposed scheme in Java on a desktop. We then ran extensive experiments and simulations to evaluate its effectiveness and efficiency in terms of computational cost and communication overhead.

The remainder of this paper is organized as follows. Section 2 shows our formalization of the system and privacy models and identification of the design goal. Next, we present the preliminaries in section 3 and the detailed design of our proposed scheme in section 4. The privacy analysis and performance evaluation are presented in sections 5 and 6, respectively. Section 7 reviews related work. Finally, section 8 concludes this paper.

2 Models and design goal

In this section, we formalize the system and privacy models for spatial crowdsourcing applications in vehicular networks, and describe our design goal.

2.1 System model

We mainly consider five entities in vehicle-based spatial crowdsourcing applications, namely the TA (Trust Authority), SC-servers (Spatial Crowdsourcing servers), RSUs (Roadside Units), SVs (Smart Vehicles; i.e., workers), and SCs (Service Consumers). Each entity can connect and communicate with each other by either wired or intermittent wireless channels, as shown in Fig. 3.

- TA: TA takes charge of the registration of SC-servers, that is, the SVs and the SCs. It also distributes the related key materials (e.g., public and private key pairs) to them.

- SC-servers: SC-servers are the spatial crowdsourcing platforms run by a private company, such as the Amazon Mechanical Turk, and have powerful computation and storage capability. These servers can accept the spatial-temporal tasks from the SCs and publish these tasks in their crowdsourcing platforms. In addition, any registered worker can accept the posted jobs, report the results, and receive rewards based on the platforms.

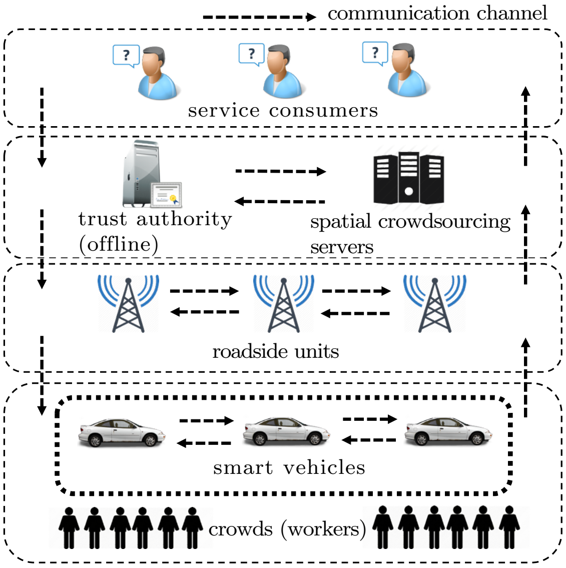


Figure 3 System model under consideration

- RSUs: RSUs are viewed as the communication nodes in our model. Being deployed in the city, they act as the middleware layer to build the connection between the workers and SC-servers. In other words, they can communicate with the nearby SVs as well as SC-servers, and deliver the data packets received from the SVs to SC-servers.
- SVs: SVs are regarded as workers who volunteer to accept and perform open tasks in the crowdsourcing platforms for some substantial rewards/incentives. Being equipped with various sensors, SVs can accomplish different task types, such as traffic monitoring and air condition sensing. In addition, the drivers of SVs can even achieve more complex tasks based on human intelligence.
- SCs: SCs are regarded as consumers of the crowdsourcing service provided by SC-servers. They can send task requests to the SP and wait for the results on a pay-per-request/use basis.

To clarify the system model, we also defined the spatial-temporal task in the vehicle-based crowdsourcing applications.

Definition 1 (Spatial-temporal task) Let $Tsk = \{tsk_1, tsk_2, \cdots\}$ be a set of spatial-temporal tasks, where $tsk_i = (id_{tsk_i}, loc_{tsk_i}, evnt_{tsk_i}, [t^s_{tsk_i}, t^e_{tsk_i}], rwd_{tsk_i})$. Each spatial-temporal task tsk_i involves a unique serial number id_{tsk_i}, specific location loc_{tsk_i}, task description $evnt_{tsk_i}$, valid time period $[t^s_{tsk_i}, t^e_{tsk_i}]$, and reward rwd_{tsk_i}.

2.2 Privacy model

As discussed in many recent research papers[4,10-12] related to spatial crowdsourcing, a significant barrier of the successful crowdsourcing applications is the privacy concern. Hence, based on the system model, we first define the trust levels of all entities from a real-world perspective, and describe the appropriate privacy requirements for protecting workers' location privacy.

- TA: TA is assumed to be fully trusted with high physical protection, and is difficult to be compromised by an attacker. For instance, TA has several servers that are ran by the official governments or trusted companies, such as GlobalSign.
- SC-servers: SC-servers are assumed to be semi-honest (i.e., honest-but-curious; they will faithfully follow procedures of protocols but may be curious and attempt to infer worker location). That is, SC-servers are one of the potential attackers.
- RSUs: RSUs are assumed to be trustable. Suppose that RSUs are not trustable, the location privacy issues for workers are nonexistent because RSUs know the exact location of any vehicle.
- SVs and SCs: Some SVs and SCs are assumed to be honest-but-curious, and they may collude with SC-servers to break the location privacy of other workers. That is, a small portion of SVs and SCs are potential attackers.

Side-information based attacks. In this study, we considered a side-information-based attack model[13]. Specifically, the attackers have the side information about some victims' location points, whose snapshots can be obtained by the attackers over a period. That is, the attackers may know the traffic information through photographs (i.e., image/video data about the vehicle going through a victim location point at an associated time instant.

The data may include the vehicle's speed and license number). In practice, the side information could be obtained easily because some victims' location points are open to observations without any restrictions. Hence, the attackers can obtain the side information directly by installing some cameras at these victim location points. With the side information, the attackers can infer the trajectories of any worker and link the real vehicle's identity to the particular worker in the crowdsourcing platform.

Privacy requirements. Based on the aforementioned trust definition, our goal is to prevent honest-but-curious attackers from extracting workers' sensitive location information while these workers voluntarily contribute the spatial-temporal reports and receive rewards. Thus, the following privacy requirements should be satisfied for the workers to be anonymous and indistinguishable.

- Anonymity: In our system, attackers only obtain a set of pseudonyms of workers instead of their real identities. That is, SC-servers can authenticate the reports provided by any worker without revealing their real identities, and these anonymous workers can finally gain their corresponding rewards.
- Untraceability: Attackers cannot easily analyze the distribution of any worker task to identify their trajectories. That is, even if some SCs or SVs collude with the SC-servers, SC-servers still cannot precisely determine the paths of workers, who accept one or several different tasks, to trace them. For instance, suppose that a worker accepts 10 spatial-temporal tasks in a day. Although SC-servers exactly know that this worker performs the corresponding tasks in these precise locations, they cannot easily reconstruct the trajectory of this worker in the day based on the collected information when untraceability is satisfied.
- Unlinkability: The probability that the attackers can link two pseudonyms should be under the privacy control of workers. That is, even if some SCs or SVs collude with the SC-servers, SC-servers still cannot clearly decide whether two pseudonyms belong to the same worker. Concretely speaking, if a worker uses a pseudonym pid_1 to accept tasks in a day and uses another pseudonym pid_2 to accept tasks the next day, SC-servers cannot easily link these two pseudonyms and identify that these two pseudonyms belong to the same worker when unlinkability is guaranteed.

A common belief that anonymity is not enough for protecting location privacy in the crowdsourcing platform is because the attackers can link two anonymous workers by analyzing the collected information, such as unique human behaviors[14]. Hence, to achieve location privacy, we consider not only the anonymity but also the untraceability and the unlinkability in the privacy model.

2.3 Design goals

Our design goal is to design a privacy-friendly spatial crowdsourcing scheme in vehicular networks, achieving the following two objectives.

- Privacy: The crowdsourcing scheme should achieve the above-mentioned privacy requirements; this is the basic goal. Concretely, the workers are able to control their privacy levels and attempt to find a trade-off between privacy and functionality (i.e., quality of protection).
- Efficiency: Although introducing privacy-preserving techniques that is used to achieve privacy preservation usually influences the efficiency of a system, the crowdsourcing scheme should also provide good user experiences (i.e., quality of experience), and the computational and communication cost should be acceptable.

3 Preliminaries

In this section, we outline the pairing technique and the Merkle tree, which serve as the building blocks of the proposed scheme.

3.1 Bilinear pairing

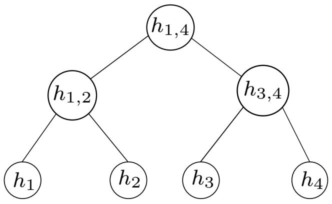
Let $\mathbb{G}$ and $\mathbb{G}_T$ be two cyclic groups of prime order q with multiplication. Further, let g be a generator of $\mathbb{G}$ and e be a bilinear map. Let $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ be a bilinear map with the following properties: i)

Bilinearity: for all $u, v \in \mathbb{G}$ and $a, b \in \mathbb{Z}_q$, we have $e(u^a, v^b) = e(u, v)^{ab}$; ii) Nondegeneracy: $e(g, g) \neq 1$; and iii) Computability: there exists an efficient algorithm to compute bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

Definition 2 (A bilinear pairing generator algorithm $Gen()$) $Gen()$ can take a security parameter ξ as input, and outputs 5-tuple parameters $(q, g, \mathbb{G}, \mathbb{G}_T, e)$.

3.2 Merkle tree

Suppose that we have m values $x_1,\cdots, x_m$, where m is a power of 2 (if m is not the power of 2, we can insert random values to satisfy the requirement). Let $H' : \{0, 1\}^* \ to \{0, 1\}^*$ be a one-way hash function. Then, the Merkle tree can be constructed using the m values $x_1,\cdots, x_m$ under the hash function H, and be represented as a balanced binary tree in which each node is associated with a hash value. There are m leaf nodes, and for each leaf node h_i, the hash value is $H'(x_i)$, where $i \in [1, m]$. In addition to leaves, other nodes in the Merkle tree are derived from its left and right childs' hash values. For instance, the hash value of the node $h_{i,i+1}$ is $H'(H'(x_i)|H'(x_{i+1}))$, where $|$ is the concatenation operation. For simplicity, we consider an example of a Merkle tree with only four values (x_1, x_2, x_3, and x_4) and construct the Merkle tree, as shown in Fig. 4.



node	hash value
h_1	$v_1{=}H(x_1)$
h_2	$v_2{=}H(x_2)$
h_3	$v_3{=}H(x_3)$
h_4	$v_4{=}H(x_4)$
$h_{1,2}$	$v_{1,2}{=}H(v_1 \vert v_2)$
$h_{3,4}$	$v_{3,4}{=}H(v_3 \vert v_4)$
$h_{1,4}$	$v_{1,4}{=}H(v_{1,2} \vert v_{3,4})$

Figure 4 Merkle tree for four values

4 Proposed privacy-friendly spatial crowdsourcing scheme

In this section, we describe our proposed spatial crowdsourcing scheme in vehicular networks. Before describing each part in our scheme, we first present a privacy metric for evaluating the privacy exposure of workers who accept various spatial-temporal tasks with different pseudonyms.

4.1 A privacy metric for spatial crowdsourcing tasks

The proposed composite metric considers three parts to measure the location privacy of a worker: the frequency of changing pseudonyms, degree of task distribution, and degree of task similarity.

4.1.1 Definition of privacy

To ensure clarity about the privacy of workers in vehicular networks, we first define the location privacy in spatial crowdsourcing applications. A general definition of privacy in Ref. [15] states privacy as the degree to which an entity cannot be linked to its real identity. Similarly, the workers' location privacy can be defined as follows in real-world spatial crowdsourcing applications.

Definition 3 (Worker's location privacy) The degree to which the spatial-temporal characteristic of a worker cannot be linked to its identities and trajectories, although the worker accepts multiple spatial-temporal tasks in the crowdsourcing platform over a contiguous series of time intervals.

4.1.2 Frequency of changing pseudonyms

If SV, as a worker in the crowdsourcing applications, does not change the pseudonym, the tasks he/she has accepted will easily be linked to his/her trajectory and used for further analysis. If a worker changes pseudonym for each spatial-temporal task, the information revealed by the tasks cannot be directly linked, and the workers can obviously obtain a higher privacy level. Thus, it is desirable that the worker alters his/her pseudonym only when it is necessary as changing pseudonyms has additional costs. Hence, we consider a k-time pseudonym of each worker in our proposed scheme, that is, a worker will alter his/her pseudonyms for k tasks in a day, where k is controlled by the worker. Then, the pseudonym duration $|T|$ is decided by k.

• Pseudonym duration. $|T|$ is defined as the time duration of the worker not changing pseudonyms, so T is a collection of contiguous time intervals starting from the first task to the k-th task. Formally, $T = \{t_j | j = 1, 2, \cdots, k-1\}$, where t_j is the time interval between two tasks.

4.1.3 Degree of task distribution

This degree evaluates a worker's untraceability, which can be measured by testing the CSR (Complete Spatial Randomness) of the spatial dataset and time deviation in both spatial and temporal dimensions. According to the definition of CSR, if the spatial dataset exhibits complete spatial randomness, no underlying structure can be obtained from it, and therefore the dataset's further analysis will gain an insignificant amount of information. That is, given that there are k location points inside a region, CSR can help measure the degree to which the locations of these task points are independent and identically distributed and uniformly distributed inside the region.

A large number of methods have been proposed to test the spatial dataset for CSR[16]. However, all these tests are basically sensitive to measuring two types of randomness: local and global randomness. Local randomness implies a type of test that is good at detecting aggregation and regularity (i.e., clustering) but not at detecting heterogeneity (i.e., spacing). In contrast, global randomness implies a type of test that is good at detecting clustering but not spacing. Hence, we choose an extension of Pollard's test flexible for recognizing both the local and global randomness.

Based on the extension of Pollard's test, by inputting the dataset of location points $\{(X, Y)|(x_1, y_1), (x_2, y_2), \cdots, (x_k, y_k)\}$, we can use the following equation to calculate the degree of task distribution.

$$d_s = \frac{C_1[kln(C_2) - C_3]}{C_4},$$

where $C_1 = 12j^2k$, $C_2 = \sum_{i=1}^{k} \frac{x_{ij}^2}{k}$, $C_3 = \sum_{i=1}^{k} \ln x_{ij}^2$, and $C_4 = (6jk + k + 1)(k - 1)$. Here, j denotes the j-th nearest neighbor (j can be 1,2,3$\cdots$), x_{ij} is the distance from the i-th point to its j-th Euclidean nearest neighbor, and k is the total number of location points due to k tasks. Then, if the degree $d_s = 1$, the location distribution of tasks satisfies complete spatial randomness. $d_s < 1$ indicates the overdispersion (spacing) of the location points and $d_s > 1$ indicates the underdispersion (clustering) of the location points.

In addition, we should consider untraceability in the temporal dimension. Suppose that the average velocity of a worker is V, and the distance (i.e., the real distance in a real-world map obtained from the state-of-the-art applications, such as Google map) between two continuous tasks is $D = Distance(A, B)$, then the average time for the worker travelling from A to B can be calculated as $t_{avg} = D/V$. If the time interval between two continuous tasks is $t_{A,B}$, the time deviation between two consecutive tasks can be calculated as $t_{\text{dev}} = |t_{A,B} - t_{\text{avg}}|$. The average time deviation is $d_t = \sum_{i=1}^{k-1} (t_{dev_i})/(k-1)$.

The degree of task distribution $degree_t$ can be measured by $(|d_s - 1| \leqslant th_s, d_t \geqslant th_t)$, where th_s and th_t are the thresholds that should be controlled by the worker.

4.1.4 Degree of task similarity

This degree is proposed to evaluate worker unlinkability, which can be measured by computing the trajectory similarity of spatial-temporal tasks under various pseudonyms. Actually, the modeling and interpreting of the trajectories based on the spatial-temporal tasks is difficult because of the sparse and diverse formats of the tasks' location information. Therefore, to quantify the behavior similarity of two trajectories, we use two general methods, the Hausdorff distance and Frechet distance in spatial and/or temporal dimensions. In other words, the similarity of two trajectories is not only affected by the geometric characteristics (i.e., shape) of trajectories but also correlates to the temporal aspect (i.e., speed). Note that, the tasks' content is not under consideration as the third dimension because we assume that the workers have similar abilities and report normalized data. For example, the photos should be normalized to the same setting before the workers report them to the SC-servers in a spatial-temporal photo-taking task.

Hausdorff distance is commonly used for trajectory matching, and the detailed algorithm is shown in Algorithm 1. Formally, Hausdorff distance can be considered as HDIST(A, B), that is, the worst-case discrepancy of one trajectory A with respect to another trajectory B. Under the metric of Euclidean distance, A is similar to B if each location point in A is close to at least one location point in B. Hausdorff distance is an elastic method, that is, the method seeks the maximum conflict rather than attempting to match each point in A to a point in B. However, this property also introduces errors and makes the similarity measurement inaccurate; thus, the Frechet distance is needed as a supplement.

Algorithm 1 HDIST(A, B)

```
Input: Location Point Set $A = (a_1, a_2, \cdots, a_n)$ and $B = (b_1, b_2, \cdots, b_m)$.
Output: Hausdorff distance from $A$ to $B$, $d_h$.
$d_h = 0$;
for each point $a_i$ in $A$ do
    $d_{shortest} = \infty$;
    for each point $b_j$ in $B$ do
        $d_{ij} = distance(a_i, b_j)$;
        if $d_{ij} < d_{shortest} then$
            $d_{shortest} = d_{ij}$;
        end if
    end for
    if $d_{shortest} > d_h$ then
        $d_h = d_{shortest}$;
    end if
end for
return $d_h$;
```

Unlike Hausdorff distance, Frechet distance can measure the similarity between two trajectories by considering both the location points and ordering of these points along the trajectories. Formally, Frechet distance can be defined as FDIST$(A(t), B(t))$, that is, the minimum length of leash necessary to connect trajectories A and B. The detailed algorithm for calculating Frechet distance is shown in Algorithm 2.

Algorithm 2 FDIST$(A(t), B(t))$

```
Input: Ordered Location Point Set $A = (a_1, a_2, \cdots, a_n)$ and $B = (b_1, b_2, \cdots, b_m)$.
Output: Frechet distance between $A$ and $B$, $d_f$.
function cal$(ca, i, j, A, B)$
    if $ca[i, j] > -1$ then
        return $ca[i, j]$;
    else if $i == 1$ and $j == 1$ then
        $ca[i, j] = distance(a_1, b_1)\}$;
    else if $i > 1$ and $j == 1$ then
        $ca[i, j] = max\{cal(ca, i-1, 1, A, B),$
            $distance(a_i, b_1)\}$;
    else if $i == 1$ and $j > 1$ then
        $ca[i, j] = max\{cal(ca, 1, j-1, A, B),$
            $distance(a_1, b_j)\}$;
    else if $i > 1$ and $j > 1$ then
        $ca[i, j] = max\{min\{cal(ca, i-1, j, A, B),$
            $cal(ca, i-1, j-1, A, B),$
            $cal(ca, i, j-1, A, B)\},$
            $distance(a_i, b_j)\}$;
    else
        $ca[i, j] = \infty$;
    end if
    return $ca[i, j]$;
Initialize the array $ca[1 \cdots n, 1 \cdots m]$ as $-1$;
return $d_f = cal(ca, n, m, A, B)$;
```

The degree of task similarity $degree_s$ can be measured by $(d_h \leqslant th_h, d_f \leqslant th_f)$, where th_h and th_f are the thresholds that should be controlled by the worker.

4.2 Privacy-friendly spatial crowdsourcing

In this section, we present our privacy-friendly crowdsourcing scheme consisting of three parts:

bootstrapping, privacy-friendly crowdsourcing, and revocation.

4.2.1 Bootstrapping

The scheme has a fundamental bootstrapping phase, in which TA generates all system parameters and registers the workers and SC-servers. Specifically, given the security parameter τ, TA runs $Gen()$ to generate the bilinear pairing parameters $(q, g, \mathbb{G}, \mathbb{G}_T, e)$. TA then selects a random element $x_{TA} \in Z_q^*$ and generates a pair of public and private keys $(pk_{TA} = g^{x_{TA}}, sk_{TA} = x_{TA})$ for itself. The public key pk_{TA} is open to all workers and SC-servers; the private key sk_{TA} is a secret that is used to issue certificates for the registration of workers and SC-servers. TA also defines two public cryptographic hash functions $H()$ and $H_0()$, where $H : \{0,1\}^* \rightarrow \mathbb{G}$ and $H_0 : \{0,1\}^* \rightarrow Z_q^*$.

- SC-server registration. An SC-server sends the registration request to TA. After verifying the identity id_{SC} of SC-server, TA selects a random element $x_{SC} \in Z_q^*$ and generates a pair of public and private keys $(pk_{SC} = g^{x_{SC}}, sk_{SC} = x_{SC})$ for the SC-server. It also assigns the server a certificate $cert_{SC}$, which can be used to confirm the validity of its identity. Generally, TA uses sk_{SC} to generate a signature $sig_{SC} = H(id_{SC}|pk_{SC})^{x_{TA}}$ on id_{SC} and pk_{SC}. TA finally outputs the certificate as a tuple $cert_{SC} = (id, pk_{SC}, sig_{SC})$, and sends the certificate $cert_{SC}$ and the corresponding private key sk_{SC} to the SC-server.

- SV/SC registration. A worker (SV) W_i or an SC C_j sends the registration request to TA. After verifying the identity id_{W_i} of the worker or the identity id_{C_j} of the SC, similar to SC-server registration, a new random element $x_{W_i} \in Z_q^*$ or $x_{C_j} \in Z_q^*$ is selected to generate a pair of public and private keys $(pk_{W_i} = g^{x_{W_i}}, sk_{W_i} = x_{W_i})$ for the worker or $(pk_{C_j} = g^{x_{C_j}}, sk_{C_j} = x_{C_j})$ for the SC. Then, a certificate will be constructed for the worker as $cert_{W_i} = (id_{W_i}, pk_{W_i}, sig_{W_i})$ or for the SC as $cert_{C_j} = (id_{C_j}, pk_{C_j}, sig_{C_j})$, where $sig_{W_i} = H(id_{W_i}|pk_{W_i})^{x_{W_i}}$ is the signature of $id_{W_j}|pk_{W_j}$ and $sig_{C_j} = H(id_{C_j}|pk_{C_j})^{x_{C_j}}$ is the signature of $id_{C_j}|pk_{C_j}$. TA finally sends the certificate $cert_{W_i}$ and private key sk_{W_i} to the worker, or sends the certificate $cert_{C_j}$ and private key sk_{C_j} to the SC.

Note that, TA is entirely offline in the process of crowdsourcing, and is only responsible for registration and revocation.

4.2.2 Privacy-friendly crowdsourcing

In our system, SC-servers run the crowdsourcing platform, and the following steps show the details of privacy-friendly crowdsourcing on this platform.

Phase 1. Task posting: A service consumer C_j first generates a spatial-temporal task $tsk_j = (id_{tsk_j}, loc_{tsk_j}, evnt_{tsk_j}, [t^s_{tsk_j}, t^e_{tsk_j}], rwd_{tsk_j})$, and computes a resignature key for this task as

$$R_{tsk_j} = pk_{SC}^{\frac{1}{x_{C_j}+H_0(id_{tsk_j})}} = g^{\frac{x_{SC}}{x_{C_j}+H_0(id_{tsk_j})}}.$$

Then, C_j submits a 6-tuple $(id_{C_j}, cert_{C_j}, tsk_j, sig_{tsk_j} = H(tsk_j|TS)^{x_{C_j}}, R_{tsk_j}, TS)$ to the SC-server, where TS is the current timestamp identified as $date|time$. After receiving the 6-tuple, the SC-server opens the certificate $cert_{C_j}$ and obtains the public key pk_{C_j} and signature sig_{C_j}. It checks $cert_{C_j}$ to verify that (id_{C_j}, pk_{C_j}) is generated by TA, and checks whether the SC is valid and his/her certificate has been revoked in the certificate revocation list CRL (detailed in Revocation section). It also checks the signature sig_{tsk_j} to validate tsk_j. If any check fails, the SC-server rejects the task; otherwise, it publishes tsk_j in its crowdsourcing platform. In addition, the SC-server chooses and stores a random element $\theta_{tsk_j} \in Z_q^*$, and publishes id_{C_j}, $cert_{C_j}$ $R_{tsk_j}^{\theta_{tsk_j}}$, and $pk_{SC}^{\theta_{tsk_j}}$ together with tsk_j as $(id_{C_j}, cert_{C_j}, R_{tsk_j}^{\theta_{tsk_j}}, pk_{SC}^{\theta_{tsk_j}}, tsk_j)$.

Phase 2. Worker preparation: A worker W_i first picks a random seed S_{W_i} and a random number $r_{W_i} \in Z_q^*$. Then, W_i generates his/her pseudonym as $pid_{W_i} = H_0(S_{W_i})$ (W_i can repeatedly use the random seed S_{W_i} to generate more pseudonyms) and blinds the pseudonym as

$$U_{W_i} = H(pid_{W_i}|date)^{r_{W_i}},$$

where $date$ is the current date indicating when the pseudonym is generated. In addition, W_i sends the authorization request $(id_{W_i}, cert_{W_i}, U_{W_i}, sig_{U_{W_i}} = H(U_{W_i}|TS)^{x_{W_i}}, TS)$ to the SC-server. Similarly, after receiving the request, the SC-server decides whether the request is legitimate by checking the certificate and signatures. If all checks pass, it calculates

$$\begin{aligned} U^*_{W_i} &= U_{W_i}^{\frac{1}{H_0(date)+x_{SC}}} \\ &= H(pid_{W_i}|date)^{\frac{r_{W_i}}{H_0(date)+x_{SC}}} \end{aligned}$$

and sends a response $(id_{SC}, cert_{SC}, U^*_{W_i}, sig_{U^*_{W_i}} = H(U^*_{W_i}|TS)^{x_{SC}}))$ to the worker. After receiving the response, W_i first checks the validity of the response according to the certificate and signatures. If the check is successful, W_i obtains, unblinds, and stores the anonymous credentials as

$$\begin{aligned} \sigma_{W_i} &= (U^*_{W_i})^{r^{-1}_{W_i}} \\ &= H(pid_{W_i}|date)^{\frac{1}{H_0(date)+x_{SC}}}. \end{aligned}$$

Phase 3. Task selection. According to the privacy metric defined earlier, worker W_i can control and measure his/her location privacy when selecting different tasks. A simple strategy is that worker W_i changes his/her pseudonym after one day and chooses at most k jobs on the crowdsourcing platform in one day. When selecting each task, W_i evaluates the privacy exposure, and decides whether the task is under his/her privacy settings. Specifically, a worker will set four thresholds (th_s, th_t, th_h, and th_f) according to the defined privacy metric. When choosing each task, W_i must first calculate d_s, d_t, d_h, and d_f, and then check whether $|d_s - 1| \leqslant th_s$, $d_t \geqslant th_t$, $d_h \leqslant th_h$, and $d_f \leqslant th_f$. If all conditions are satisfied, the task can be accepted by the worker. Otherwise, the task is rejected.

After choosing a suitable task tsk_j, worker W_i verifies whether task tsk_j really originates from C_j by checking

$$\begin{aligned} & e(pk_{C_j} \cdot g^{H_0(id_{tsk_j})}, R_{tsk_j}^{\theta_{tsk_j}}) = e(pk_{SC}^{\theta_{tsk_j}}, g). \\ => \quad & e(g^{x_{C_j}+H_0(id_{tsk_j})}, g^{\frac{x_{SC}\theta_{tsk_j}}{x_{C_j}+H_0(id_{tsk_j})}}) \\ = \; & e(g^{x_{SC}\theta_{tsk_j}}, g). \\ => \quad & e(g,g)^{x_{SC}\theta_{tsk_j}} = e(g,g)^{x_{SC}\theta_{tsk_j}}. \end{aligned}$$

Then, worker W_i performs the task and obtains the result $res_{tsk_j} = id_{tsk_j}|data$ if the check is successful.

Phase 4. Anonymous data reporting. To report the result for tsk_j, W_i first picks a random unique serial number s_{num}, random seed α, and random number $v_{W_i} \in Z_q^*$. Then, W_i generates a rewarding token $token_{tsk_j} = H_0(\alpha)$ and blinds the token as $\beta = H(token_{tsk_j}|id_{tsk_j})^{v_{W_i}}$. Next, W_i chooses a random element $z_{W_i} \in Z_q^*$ and encrypts $Msg = res_{tsk_j}|pid_{W_i}|\sigma_{W_i}|date$ by using the SC-server's public key as

$$Msg_{auth} = (H(g^{z_{W_i}}, pk_{SC}^{z_{W_i}}) \oplus Msg, g^{z_{W_i}}).$$

Eventually, W_i reports a 3-tuple $(s_{num}, Msg_{auth}, \beta)$ to the SC-server. After receiving the 3-tuple, the SC-server uses its private key to decrypt Msg_{auth} as

$$Msg = H((g^{z_{W_i}})^{x_{SC}}, g^{z_{W_i}}) \oplus Msg_{auth},$$

and accepts the report if

$$e(g^{H_0(date)} \cdot pk_{SC}, \sigma_{W_i}) = e(g, H(pid_{W_i}|date)).$$

Phase 5. Anonymous rewarding. C_j checks the reports for task tsk_j on the crowdsourcing platform. If C_j adopts the report from and agrees to reward W_i, the SC-server signs the token by using its private key and θ_{tsk_j} as $\delta_{W_i|tsk_j} = \beta^{x_{SC}\theta_{tsk_j}}$. Then, the SC-server publishes $(s_{num}, \delta_{W_i|tsk_j})$ on the crowdsourcing platform. Worker W_i matches s_{num} on the platform with local serial numbers to download his/her token $\delta_{W_i|tsk_j}$ and he/she unblinds the token as

$$\begin{aligned} \delta^*_{W_i|tsk_j} &= \delta_{W_i|tsk_j}^{v^{-1}_{W_i}} \\ &= H(token_{tsk_j}|id_{tsk_j})^{x_{SC}\theta_{tsk_j}}. \end{aligned}$$

To gain the payment of tsk_j, W_i sends the rewarding request $(id_{tsk_j}, \alpha, \delta^*_{W_i|tsk_j})$ to the SC-server, which verifies the request as

$$e(\delta^*_{W_i|tsk_j}, g) = e(pk_{SC}^{\theta_{tsk_j}}, H(H_0(\alpha)|id_{tsk_j})).$$

If the request passes the verification tests, the reward is sent back. To prevent repeated rewarding requests, α is stored in a temporary list $List_{tsk_j}$, and the request is first searched in $List_{tsk_j}$. If the token has been used before, it must exist in $List_{tsk_j}$ and the repeated rewarding request will be aborted.

Phase 6. Task closing. Finally, SC C_j chooses to close the task on the crowdsourcing platform, and the SC-server deletes all the information related to the task.

4.2.3 Revocation

TA can construct a Merkle tree as the certification revocation list CRL shown in Fig. 4, where the values should be the certificates of the revoked workers, such as $x_1 = cert_{C_1}$, $x_2 = cert_{C_2}$, $x_3 = cert_{C_3}$, and $x_4 = cert_{C_4}$. The root of the Merkle tree should be signed by TA as sig_{root}. TA's signature can guarantee the tree's integrity and authenticity. Then, the Merkle tree is outsourced to the SC-servers, and the SC-servers can traverse the Merkle tree to verify whether a certificate has been revoked or not. For instance, based on the Merkle tree shown in Fig. 4, the SC-server can verify certificate $cert_{C_j}$ by path $(h_2, h_{3,4}, h_{1,4})$ by first computing $\bar{h}_1 = H'(cert_{C_1})$, $\bar{h}_{1,2} = H(\bar{h}_1|h_2)$ and $\bar{h}_{1_4} = H(\bar{h}_{1,2}|h_{3,4})$. Next, the SC-server checks whether $\bar{h}_{1,4} = h_{1,4}$. If the check passes, the certificate has been revoked.

5 Privacy analysis

In this section, we discuss privacy properties of the proposed scheme. In particular, following our design goal, we will focus on analyzing how the proposed scheme is privacy-friendly in terms of anonymity, untraceability and unlinkability.

The proposed scheme achieves anonymity. According to our proposed scheme, a worker is guaranteed anonymity in two phases of privacy-friendly crowdsourcing: data reporting and rewarding. The worker is appropriately anonymously authenticated in these two phases because of the reliance on the generated blinded pseudonyms at the worker-preparation phase and the created blinded tokens at the anonymous-data-reporting phase. The partially blind signature algorithm used in our proposed scheme is similar to the algorithm proposed by Zhang et al.[17], in whose study the security is based on the computational Diffie–Hellman problem. In the worker-preparation phase, although the SC-server can authenticate the workers W_i based on their certificate and sign the pseudonyms provided by W_i, it cannot distinguish the pseudonyms in the anonymous-data-reporting phase because each pseudonym is blinded by a random element r_{W_i}. In the anonymous-data-reporting phase, while the SC-server can authenticate pseudonyms based on its signature and sign the token provided by W_i, it cannot differentiate between the tokens in the anonymous-rewarding phase because each token is blinded by a random element v_{W_i}. However, while the pseudonyms and tokens do not leak information, other side channels may be used to break the anonymity property of the scheme. A special case is that the IP address of an SV is visible when the worker submits the data/retrieves a reward to/from the SC-servers. To deal with this issue, some additional mechanisms, such as the TOR network, are required to ensure that a network connection remains anonymous.

The proposed scheme achieves untraceability and unlinkability. According to our proposed scheme, the worker guarantees untraceability and unlinkability based on the proposed privacy metric: the frequency of changing a pseudonym, degree of task distribution, and degree of task similarity. Suppose that a worker applies for multiple pseudonyms once in the worker-preparation phase and changes his/her pseudonym at every task $(k = 1)$, the worker is considered to have the highest untraceability and unlinkability because it is impossible for the SC-servers to perform the trajectory analysis/mining with only one valid data. However, the untraceability and unlinkability is affected by the following parameters: $k, degree_t = \{d_s, d_t\}$, and $degree_s = \{d_h, d_f\}$.

To measure the privacy exposure, we use a real taxi-trace dataset[9], which contains GPS coordinates of more than 533 taxis collected in 20 days in San Francisco, USA. Most of the coordinate-update frequencies of the taxi vary from 30 to 60 s. Owing to the lack of a real spatial crowdsourcing dataset, we simulate the crowdsourcing procedure by using the taxi-trace dataset. We consider that a taxi changes its pseudonym every 24 h by assuming that a taxi accepts a crowdsourcing job every 1, 2, and 4 h, corresponding to $k = 24, 12$, and 6, and sample the spatial-temporal points of two taxi's trajectories in 3 days. The results are shown in Figs. 5 and 6.

As k is decided, the degree of task distribution $degree_t$ is controlled by d_s and d_t in both spatial and temporal dimensions. It is evident from Fig. 5(a) that k influences the complete spatial randomness d_s of spatial tasks. When a worker has accepted more tasks, less spatial randomness of the worker implies that more information has been revealed to help an attacker trace the worker. Moreover, it is easy to find that the values of d_s are much larger than 1, even though the worker only takes almost six tasks (4 h/task) in one day. In other words, the worker needs to carefully choose suitable tasks to avoid disclosing more information. As shown in Fig. 5 (b), k also affects the time deviation d_t. In fact, as the time interval between two consecutive tasks is larger than 1 h in our simulation, the time correlation between two tasks is not obvious ($\geqslant$ 50 min). However, a tricky issue, in which the traffic condition negatively impacts the simulation results, appeared when we analyze the task distribution in the temporal dimension. This is because we estimate the speed of workers based on the distance and time duration obtained from Google map. The real-time data may result in inaccurate simulation values.

Figs. 6 (a) and (b) illustrate the degree of task similarity measured through Hausdorff and Frechet distances. Evaluation of the degree of task similarity is difficult because of the lack of real spatial-temporal task datasets. Although we simulate the crowdsourcing process based on the taxi-trace dataset, the results may not be precise. Nevertheless, the simulation still demonstrate a phenomenon, in which it is very difficult to link two pseudonyms of a taxi, while the attackers merely have a discrete spatial-temporal dataset. In another phenomenon, fewer tasks do not imply more difficulties in performing an attack. It seems to be easier for attackers to link two trajectories when the worker has accepted merely a small portion of sensitive tasks. Moreover, the workers are supposed to measure the degree of task similarity by using both Hausdorff and Frechet distances because sometimes only spatial correlation can reveal more information to the attackers without the consideration of time effects.

6 Performance evaluation

This section shows our analysis of the computational costs and communication overheads of the proposed scheme. Specifically, we implement our scheme and evaluate the performance of each part in our scheme. Our experiment environment is a desktop with 3.1 GHz processor, 8 GB RAM, and Window 7 platform. The proposed scheme is implemented using Java. For the bilinear pairing operations, we relied on the Java pairing-based cryptography library and chose Type A pairings. We also use SHA-512 for the cryptographic hash functions and define the security parameter $\tau = 512$.

6.1 Computational costs

We identify the major time-consuming operations for the onlined privacy-friendly crowdsourcing in the proposed scheme, as shown in Tab. 1. T_{exp} and T_{mul} denote the time cost of exponential and multiplicative operations in $\mathbb{G}$ and $\mathbb{G}_T$, respectively. The symbols T_{pair} and T_{hash} represent the time cost of bilinear pairing operation and cryptographic operation, respectively.

Table 1 Computation complexity

phases	parties	cost
phase.1	SC	$2T_{\text{exp}} + 2T_{\text{hash}}$
	SC-server	$2T_{\text{exp}} + 4T_{\text{pair}} + 2T_{\text{hash}}$
phase.2	SV	$5T_{\text{hash}} + 3T_{\text{exp}} + 4T_{\text{pair}}$
	SC-server	$2T_{\text{hash}} + 2T_{\text{exp}} + 4T_{\text{pair}}$
phase.3	SV	$T_{\text{hash}} + T_{\text{exp}} + T_{\text{mul}} + 2T_{\text{pair}}$
phase.4	SV	$3T_{\text{hash}} + 3T_{\text{exp}}$
	SC-server	$2T_{\text{exp}} + 3T_{\text{hash}} + 2T_{\text{pair}} + T_{\text{mul}}$
phase.5	SV	T_{exp}
	SC-server	$T_{\text{exp}} + 2T_{\text{hash}} + 2T_{\text{pair}}$

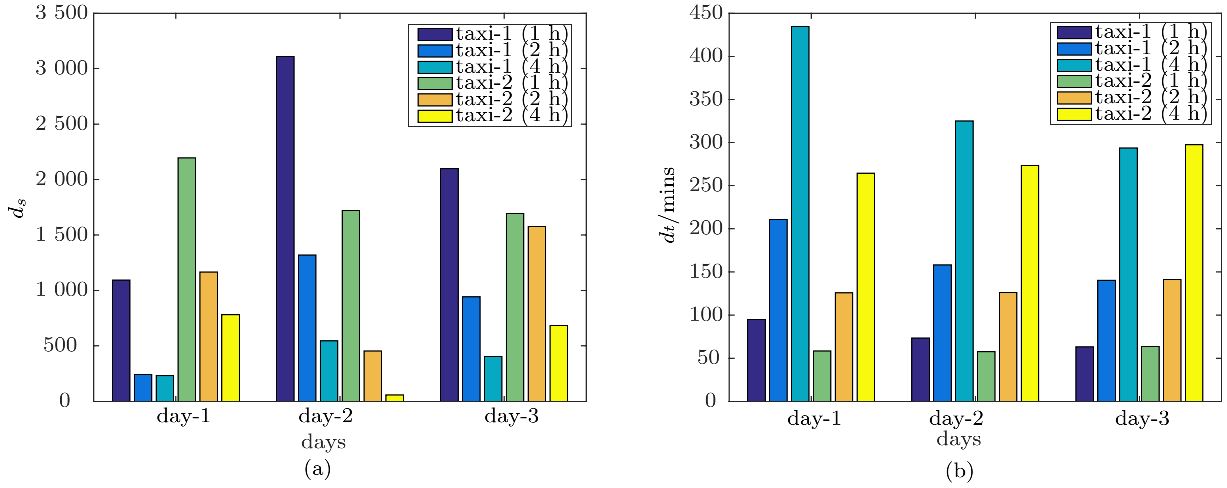


Figure 5 Degree of task distribution. (a) Spatial dimension d_s; (b) temporal dimension d_t

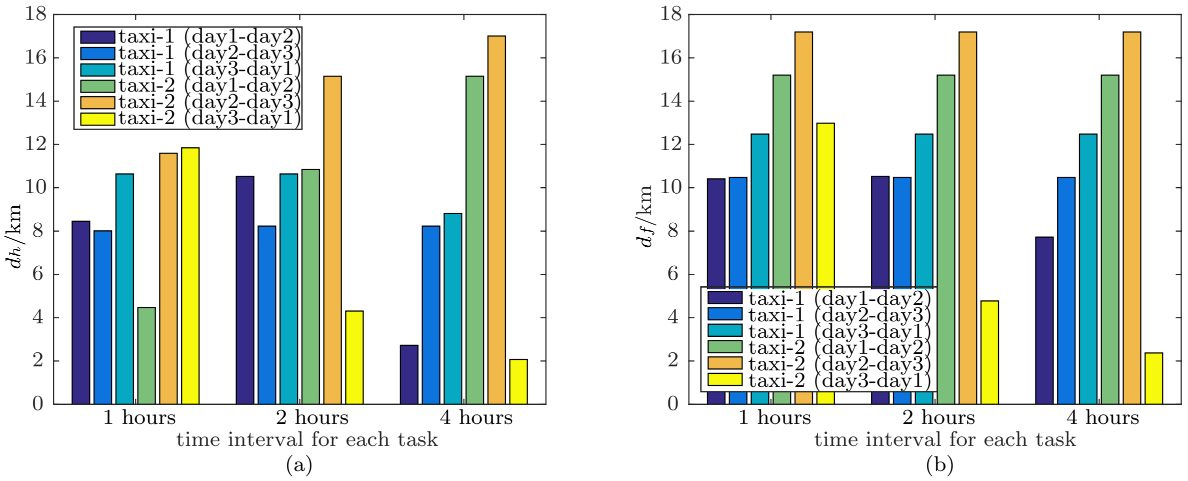


Figure 6 Degree of task similarity. (a) Hausdorff distance d_h; (b) Frechet distance d_f

For simplicity, each entity's computational cost will be measured separately. Furthermore, as the bootstrapping can be completed offline and does not affect the efficiency of a spatial crowdsourcing system, we only evaluate the online computational cost in our proposed scheme. Fig. 7 shows the results.

In our system, SC only participates in task posting. The average computation latency of submitting a crowdsourcing task to the SC-servers is almost 15 ms at SC side, while the average computation latency of accepting and publishing a job is 46 ms at the SC-server side. In the worker-preparation, workers must submit a request to SC-server and gain pseudonyms. The average running time is almost 58 ms at the SV side and 47 ms at the SC-server side. This is because the computational cost of a worker is larger than that of an SC-server as the worker must generate pseudonyms in advance. When selecting a task, the worker should perform the verification operation, which costs almost 24 ms. In the anonymous-data-reporting phase, the SV uploads anonymous credential and data, while the SC-server performs anonymous authentication. The average running time is 26 ms at the SV side and 33 ms at the SC-server side. Finally, the worker sends the rewarding token to the SC-server and gains the rewards if the token passes the anonymous verification. The computational cost of this phase is 7 and 15 ms at SV and SC-server sides, respectively.

For the revocation operation, creating the Merkle tree for the revoked certificates needs efficient hashing operations and one signature by the TA. A Merkle tree of height h supports at most 2^h certificates. The computation of the root node of this tree requires 2^h noncryptographic hashing operations and that of the interior nodes requires $2^h - 1$ noncryptographic hashing operations. To verify whether a certificate is revoked, the SC-server should perform $h+1$ hashing operations. Several schemes have been proposed to improve the efficiency and effectiveness of the Merkle tree regarding computation and storage. For instance, Dahlberg et al.[18] proposed an efficient sparse Merkle tree structure to balance the cost of space and time.

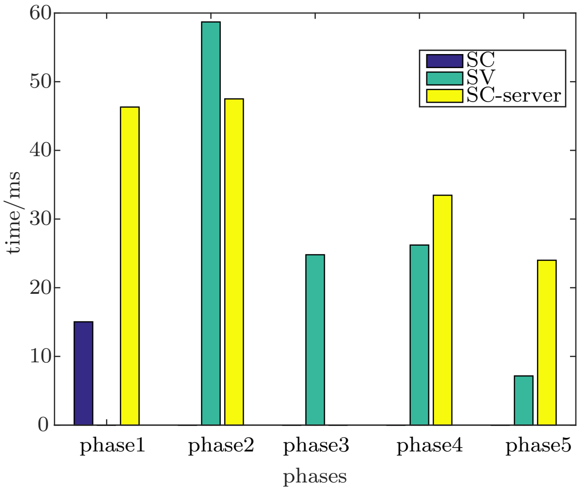


Figure 7 Computational cost of each phase

6.2 Communication overheads

In our proposed scheme, the communication overhead of the large-sized ciphertext is a major concern. Therefore, we focus on analyzing the communication overhead when the transmission data is encrypted using cryptographic techniques in the wireless communication. As we measured in the simulation, when an SV uploads his/her message, the message size is less than 10 KB, which is acceptable. Note that, we only evaluate the message size without data because the data size could be variable as a result of different task types. We also considered that the anonymous-data-reporting and anonymous-rewarding phases are two bottlenecks for the crowdsourcing system, and there may be a large number of concurrent requests from workers. Hence, when implementing our scheme, we also spawn multiple threads on the SC-server, each thread corresponding to a single worker submitting the report and sending the rewarding request. Our findings show that our scheme scales well with the increased number of SVs. The SC-server can handle up to 200 concurrent SV requests/s for data reporting, with a response time of 100 ms for each request. Moreover, the SC-server can handle 250 rewarding requests/s with a response time of 88 ms for each request.

7 Related works

In recent years, spatial crowdsourcing has attracted the interest from the research community (see Refs. [19-22]). Generally, there are two types of spatial crowdsourcing schemes based on different data sources (i.e., workers), namely: mobile device users[23] and vehicles[24]. With the rapid development of VANET and communication technology, vehicle-based crowdsourcing applications have become increasingly realistic and many crowdsourcing applications have been proposed, for example, real-time navigation[25], air-quality sensing[11,12], and traffic monitoring[26].

Many research topics have been studied in the area of spatial crowdsourcing, such as how to generate the strategy for task assignment/allocation[20], how to detect the truth from the crowdsourced results[21], and how to design a proper incentive mechanism for the workers[22]. Although spatial crowdsourcing applications can lead to significant benefits to SCs, there are fundamental security and privacy issues that must be examined[10,27], such as how to protect the location privacy of workers[3-7].

To protect the worker's location privacy in spatial crowdsourcing, To et al.[4] introduced a secure framework for spatial crowdsourcing based on differential privacy and developed an interactive visualization and tuning toolbox for privacy-preserving spatial crowdsourcing in a subsequent study[6]. Building on To's framework, Shen et al.[7] proposed a new framework under the honest-but-curious model. Moreover, Hu et al.[3] developed an approach to protect location privacy in spatial crowdsourcing based on k-anonymity. From another perspective, Gong et al.[5] proposed a privacy-preserving task recommendation scheme based on location and other sensitive information. The scheme is designed to recommend suitable crowdsourcing tasks to mobile workers who are concerned about their privacy. Furthermore, Ni et al.[25] designed an anonymous crowdsourcing system for navigation applications based on the group signature technique.

Our proposed scheme differs from existing literatures in that we attempted to address the location privacy issues of spatial crowdsourcing from a different viewpoint so that the sparse spatial-temporal tasks can still be linked to the workers' trajectories; none of the existing schemes can deal with this issue except for the proposed scheme. In addition, we also integrated an efficient anonymous technique with a new privacy metric to offer a better crowdsourcing service in a privacy-friendly manner.

8 Conclusion

In this paper, we proposed a privacy-friendly spatial crowdsourcing scheme in vehicular networks. The proposed scheme can not only allow the workers to accept crowdsourcing tasks, report the results, and gain rewards anonymously but also allows them to control their privacy leakage according to the proposed privacy metric. Detailed privacy analyses show that the proposed scheme is privacy-friendly under our defined privacy model. In addition, extensive performance analyses and experiments were conducted, and the results indicate that the proposed scheme is efficient in both computational costs and communication overheads. In future, we will consider a more sophisticated problem about how to balance the privacy issue and the task's income so that the location privacy can be guaranteed while the worker can gain a better benefit by choosing appropriate tasks.

References

[1] S. H. Kim, Y. Lu, G. Constantinou, et al. Mediaq: mobile multimedia management system [C]//Proceedings of the 5th ACM Multimedia Systems Conference, 2014: 224-235.

[2] X. Wang, X. Zheng, Q. Zhang, et al. Crowdsourcing in ITS: the state of the work and the networking [J]. IEEE transactions on intelligent transportation systems, 2016, 17(6): 1596-1605.

[3] J. Hu, L. Huang, L. Li, et al. Protecting location privacy in spatial crowdsourcing [C]//Asia-Pacific Web Conference Web Technologies and Applications, 2015: 113-124.

[4] H. To, G. Ghinita, C. Shahabi. A framework for protecting worker location privacy in spatial crowdsourcing [J]. Proceedings of the VLDB endowment, 2014, 7(10): 919-930.

[5] Y. Gong, Y. Guo, Y. Fang. A privacy-preserving task recommendation framework for mobile crowdsourcing [C]//IEEE Global Communications Conference (GLOBECOM), 2014: 588-593.

[6] H. To, G. Ghinita, C. Shahabi. PrivGeoCrowd: a toolbox for studying private spatial crowdsourcing [C]//IEEE 31st International Conference on Data Engineering (ICDE), 2015: 1404-1407.

[7] Y. Shen, L. Huang, L. Li, et al. Towards preserving worker location privacy in spatial crowdsourcing [C]//IEEE Global Communications Conference (GLOBECOM), 2015: 1-6.

[8] N. Yang, P. Yu. Efficient Hidden trajectory reconstruction from sparse data [C]//Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, 2016: 821-830.

[9] M. Piorkowski, N. Sarafijanovic-Djukic, M. Grossglauser. A parsimonious model of mobile partitioned networks with clustering [C]//IEEE Communication Systems and Networks and Workshops, 2009: 1-10.

[10] Y. Zhao, Q. Han. Spatial crowdsourcing: current state and future directions [J]. IEEE communications magazine, 2016, 54(7): 102-107.

[11] C. Xu, R. Lu, H. Wang, et al. PAVS: a new privacy-preserving data aggregation scheme for vehicle sensing systems [J]. Sensors, 2017, 17(3): 500.

[12] H. Hu, R. Lu, C. Huang, et al. TripSense: a trust-based vehicular platoon crowdsensing scheme with privacy preservation in VANETs [J]. Sensors, 2016, 16(6): 803.

[13] C. Y. T. Ma, D. K. Y. Yau, N. K. Yip, et al. Privacy vulnerability of published anonymous mobility traces [J]. IEEE/ACM transactions on networking (TON), 2013, 21(3): 720-733.

[14] Y. A. De Montjoye, C. A. Hidalgo, M. Verleysen, et al.

Unique in the crowd: the privacy bounds of human mobility [J]. Scientific reports, 2013, 3: 1376.

[15] G. P. Corser, H. Fu, A. Banihani. Evaluating location privacy in vehicular communications and applications [J]. IEEE transactions on intelligent transportation systems, 2016, 17(9): 2658-2667.

[16] M. J. Fortin, M. R. Dale. Spatial analysis: a guide for ecologists [M]. Cambridge: Cambridge University Press, 2005.

[17] F. Zhang, R. Safavi-Naini, W. Susilo. Efficient verifiably encrypted signature and partially blind signature from bilinear pairings [C]//International Conference on Cryptology, 2003: 191-204.

[18] R. Dahlberg, T. Pulls, R. Peeters. Efficient sparse Merkle trees: caching strategies and secure (non-) membership proofs [C]//Proceedings of the 21st Nordic Workshop on Secure Computer Systems (NORDSEC 2016). 2016.

[19] L. Kazemi, C. Shahabi, L. Chen. Geotrucrowd: trustworthy query answering with spatial crowdsourcing [C]//Proceedings of the 21st ACM sigspatial international conference on advances in geographic information systems, 2013: 314-323.

[20] Y. Tong, J. She, B. Ding, et al. Online mobile micro-task allocation in spatial crowdsourcing [C]// IEEE 32nd International Conference on Data Engineering (ICDE), 2016: 49-60.

[21] R. W. Ouyang, M. Srivastava, A. Toniolo, et al. Truth discovery in crowdsourced detection of spatial events [J]. IEEE transactions on knowledge and data engineering, 2016, 28(4): 1047-1060.

[22] X. Zhang, G. Xue, R. Yu, et al. Truthful incentive mechanisms for crowdsourcing [C]//IEEE Conference on Computer Communications (INFOCOM), 2015: 2830-2838.

[23] J. Ren, Y. Zhang, K. Zhang, et al. Exploiting mobile crowdsourcing for pervasive cloud services: challenges and solutions [J]. IEEE communications magazine, 2015, 53(3): 98-105.

[24] D. Wu, Y. Zhang, L. Bao, et al. Location-based crowdsourcing for vehicular communication in hybrid networks [J]. IEEE transactions on intelligent transportation systems, 2013, 14(2): 837-846.

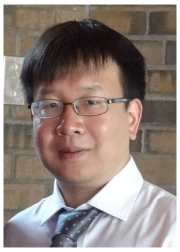
[25] J. Ni, X. Lin, K. Zhang, et al. Privacy-preserving real-time navigation system using vehicular crowdsourcing [C]//IEEE 84th Vehicular Technology Conference (VTC-Fall), 2016: 1-5.

[26] R. Lu, X. Lin, Z. Shi, et al. A lightweight conditional privacy-preservation protocol for vehicular traffic-monitoring systems [J]. IEEE intelligent systems, 2013, 28(3): 62-65.

[27] K. Yang, K. Zhang, J. Ren, et al. Security and privacy in mobile crowdsourcing networks: challenges and opportunities [J]. IEEE communications magazine, 2015, 53(8): 75-81.

About the authors

Cheng Huang [corresponding author] received his B.Eng. and M.Eng. from Xidian University, China, in 2013 and 2016 respectively, and was a project officer with the INFINITUS laboratory at the School of Electrical and Electronic Engineering, Nanyang Technological University till July 2016. Since September 2016, he has been a Ph.D. candidate with the Department of Electrical and Computer Engineering, University of Waterloo, ON, Canada. His research interests is in the areas of applied cryptography, cyber security and privacy. (Email: c225huan@uwaterloo.ca)



Rongxing Lu (S'09-M'10-SM'15) has been an assistant professor at the Faculty of Computer Science, University of New Brunswick (UNB), Canada, since August 2016. Before that, he worked as an assistant professor at the School of Electrical and Electronic Engineering, Nanyang Technological University (NTU), Singapore from April 2013 to August 2016. Rongxing Lu worked as a postdoctoral fellow at the University of Waterloo from May 2012 to April 2013. He was awarded the most prestigious "Governor General's Gold Medal", when he received his Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, Canada, in 2012; and won the 8th IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award, in 2013. He is presently a senior member of IEEE Communications Society. His research interests include applied cryptography, privacy enhancing technologies, and IoT-big data security and privacy. Dr. Lu currently serves as the Secretary of IEEE ComSoc CIS-TC. (Email: rlu1@unb.ca)

Hui Zhu (M'13) received the B.Sc. and Ph.D. degrees from Xidian University, Xi'an, China, in 2003 and 2009, respectively, and the M.Sc. degree from Wuhan University, Wuhan, China, in 2005. In 2013, he was with School of Electrical and Electronics Engineering, Nanyang Technological University as a Research Fellow. Since 2016, he has been the professor in the School of Cyber Engineering, Xidian University, China. His research interests include the areas of applied cryptography, data security and privacy. (Email: xdzhuhui@gmail.com)