Computers & Security 95 (2020) 101850

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose

A distributed location obfuscation method for online route planning

Padraig Corcoran[a,*], Peter Mooney[b], Andrei Gagarin[c]

[a] School of Computer Science & Informatics, Cardiff University, Wales, UK
[b] Department of Computer Science, Maynooth University, Maynooth, Ireland
[c] School of Mathematics, Cardiff University, Wales, UK

ARTICLE INFO

Article history:
Received 25 September 2018
Revised 15 February 2020
Accepted 20 April 2020
Available online 3 May 2020

2010 MSC:
00-01
99-00

Keywords:
Privacy
Online route planning
Distributed computing
Street network
Location obfuscation

ABSTRACT

A novel location obfuscation method for online route planning is proposed which is robust to privacy inferences by the service provider regarding route source and destination. This is achieved by performing the task of route computation in a distributed manner. Specifically, the client decomposes the required route into a sequence of shorter routes between intermediate locations. These routes are subsequently requested from independent online route planners with the results being integrated by the client to give the route originally required. Robustness to privacy inferences is a consequence of the fact that, without significant coordination and sharing of information, an individual online route planner cannot infer with high probability the true route source or destination. An evaluation of the proposed method is performed in the context of route planning within the street network of Boston. This evaluation demonstrates that the proposed method offers robustness to privacy inferences while exhibiting a reasonable reduction in quality of service.

1. Introduction

The ubiquity of location aware mobile devices has resulted in the birth of a new type of online service known as a location-based service (LBS) where the service provided is a function of the client's location. Online route planners represent one of the most popular types of LBS where the client requests the specification of a route from a service provider (Mooney and Corcoran, 2012). Such requests are usually made subject to constraints such as the route must follow the street network or use a particular transportation mode. Unlike their offline counterparts, online route planners typically take advantage of real time information related to traffic and weather conditions when recommending routes to clients (Vicente et al., 2011). The wide spread usage of online route planners has raised concerns regarding the potential for privacy inference attacks whereby an attacker infers private information relating to the client (Krumm, 2007). If a client requests such a service, an attacker in the form of a service provider or someone eavesdropping on the communication could potentially infer much private information regarded the client (Lee et al., 2009). If one assumes the use of a secure communication protocol, such as SSH which exploits advances in encryption, the threat posed by an attacker eavesdropping on the communication can be considered minimal. Therefore the most significant threat is that posed by an attacker in the form of a service provider, and this represents the focus of this paper.

The most important aspects of a route which may be used by a service provider to infer private information are the corresponding source and destination locations. To illustrate this consider the case where a client requests a route to a HIV clinic. Using this information a service provider could infer very sensitive information regarding potential health conditions of the client. Furthermore, the service provider could infer the source location to be the home of the client. Another aspect of a route which may be used by a service provider to infer private information is the route constraints. For example, if a client consistently requests routes subject to the constraint that the transportation mode is public transportation, the service provider could infer that the client does not own a vehicle.

Given the above concerns regarding inferences of private information, it is important to many clients that their usage of online route planners is robust to such inferences. Existing methods for online route planning are undistributed in nature. That is, the client requests the specification of a route from a single service provider. In their basic form, undistributed methods are not robust to inferences by the service provider regarding route source and destination; this information is explicitly communicated by the client when making a request. To overcome this

[*] Corresponding author.
E-mail address: corcoranp@cardiff.ac.uk (P. Corcoran).

https://doi.org/10.1016/j.cose.2020.101850
0167-4048/© 2020 Elsevier Ltd. All rights reserved.

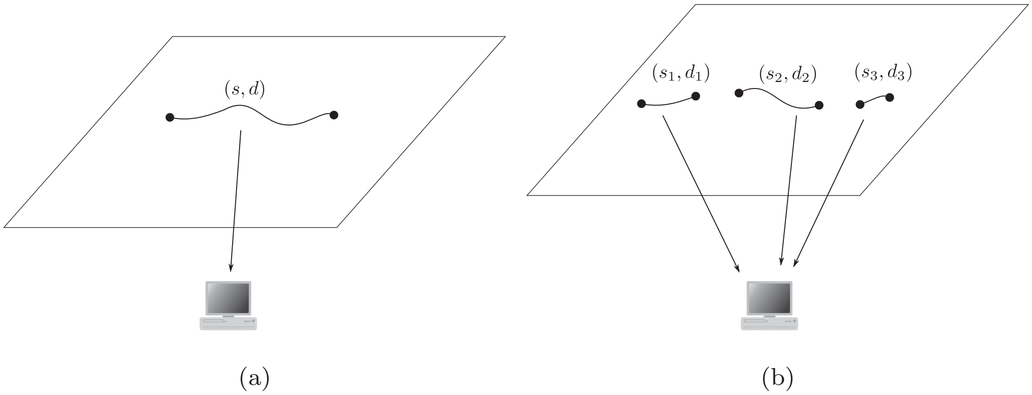

Fig. 1. In (a) the client does not employ any location obfuscation method and requests the route (s, d) from a single service provider. In (b) the client employs the proposed location obfuscation method where they decompose the route (s, d) into a sequence of 3 shorter routes (s_1, d_1), (s_2, d_2) and (s_3, d_3) such that $s_1 = s$, $d_3 = d$, $d_1 = s_2$ and $d_2 = s_3$. Each of these routes are requested from an independent online route planner.

limitation, many methods for online route planning have been proposed which achieve the robustness in question by adding an additional layer of location abstraction between the client and the service provider. For example, a commonly employed location abstraction method involves indirectly submitting a request to a service provider via a trusted anonymizer such as a virtual private network (VPN) (Rodden et al., 2002). If successful, the service provider is unable to infer the identity of the client and, in turn, cannot link/attribute any private information to them.

We propose a novel location obfuscation method for online route planning which achieves robustness to privacy inferences regarding route source and destination by performing the task of route computation in a distributed manner. To illustrate this method consider the situation where a client requires a route from a source location s to a destination location d satisfying given constraints. We denote any such route as (s, d) where the notation (.,.) denotes a route between two locations satisfying the constraints in question. When not employing any location obfuscation method, the client requests (s, d) from a single service provider. This is illustrated in Fig. 1(a). When employing the proposed location obfuscation method, the task of route computation is performed in a distributed manner. Specifically, the client decomposes the route (s, d) into a sequence of n shorter routes $\{(s_1, d_1) \ldots (s_n, d_n)\}$ such that $s_1 = s$, $d_n = d$ and $d_i = s_{i+1}$ for $i = 1 \ldots n-1$. These routes are subsequently requested from independent online route planners with the results being integrated by the client to give the route originally required. This method is illustrated in Fig. 1(b). We prove that this method is robust to privacy inferences whereby, without significant coordination and sharing of information, an online route planner cannot infer with high probability the true route source or destination. Furthermore, we demonstrate that the use of this method does not result in a significant reduction in quality of service.

The layout of this paper is as follows. In Section 2 we review existing methods for online route planning which are robust to privacy inferences. In Section 3 the proposed method for online route planning is described. Section 4 presents an evaluation of this method with respect to privacy inference robustness, time and communication complexity, and quality of service. Finally, in Section 5 we draw conclusions from this work and discuss possible future research directions.

2. Related works

In this section we review existing methods for online route planning which are robust to privacy inferences. These methods achieve the desired robustness by either employing established techniques, such as data encryption, which generalize to a wide spectrum of applications, including online route planning, and/or exploiting the specific structure of the route planning problem.

Mouratidis and Yiu (2012) and Xi et al. (2014) proposed methods employing Private Information Retrieval (PIR) which is a general protocol allowing a client to query a database located on a server without the query being revealed to the server. However this approach requires full cooperation of the service provider whereby they support this protocol. Buchanan et al. (2013) proposed a method which involves submitting an additional set of distinct requests to the service provider along with the true request. If the service provider cannot differentiate between the additional and true requests robustness is achieved. However this method requires the user to propose additional requests which are plausible and sufficiently different to the true request. Furthermore, submitting a sufficient number of additional requests introduces significant communication and path computation overhead. In a related work Lee et al. (2009) proposed to reduce this overhead using a number of optimization techniques.

Interacting with a LBS (location-based service) indirectly via a trusted anonymizer is a general approach to achieving robustness to privacy inferences which generalizes to online route planning (Luo and Yang, 2017). An anonymizer can be used to achieve $\mathcal{K}$-anonymity by constructing a $\mathcal{K}$-Anonymizing Spatial Region ($\mathcal{K}$-ASR) which contains $\mathcal{K} - 1$ other clients and submitting requests at the resolution of this region (Gruteser and Grunwald, 2003). However such approaches are vulnerable if the anonymizer is compromised (Ghinita et al., 2007). To overcome this limitation a number of solutions have been proposed which eliminate the need for a centralized anonymizer by using a distributed computing paradigm. Chow et al. (2006) proposed a peer-to-peer (P2P) method which does not require a centralized server and constructs $\mathcal{K}$-ASRs by considering groups of clients within close spatial proximity to each other. Ghinita et al. (2007b) proposed a similar model which constructs $\mathcal{K}$-ASRs using an overlay network which resembles a B$^+$-tree. In a related work, Xu et al., 2018 proposed a model whereby clients are clustered into groups based on trajectory similarity, and these groups are in turn used to achieve $\mathcal{K}$-anonymity. Similarly, Shokri et al. (2014) proposed a method where clients collaborate and share information, thus minimizing the number of requests made to the service provider. As identified by Zhong and Hengartner (2008), one of the issues with these distributed models is that each client must trust all other clients. To overcome this limitation, the authors proposed a method based on encryption which limits the information exchanged between clients. A number of authors have proposed methods for constructing $\mathcal{K}$-ASRs in situations where a client's location is constrained to lie in a street network (Chow et al., 2011; Mouratidis and Yiu, 2010).

As noted by Dorfmeister et al. (2015), $\mathcal{K}$-anonymity is not a sufficient condition for achieving robustness to inferences regarding the source and destination of a route. If all $\mathcal{K}$ destinations are contained within a region which corresponds to a single realistic destination, such as the grounds of a particular hospital, an attacker can directly infer the destination. The same argument holds for the source location. In this work we refer to regions corresponding to a single realistic source or destination as *semantic regions*. Note that a location may not lie in a semantic region. Examples of such locations include the center of a lake or a motorway. To overcome the above limitation of $\mathcal{K}$-anonymity, the concept of l-diversity has been proposed and is achieved when the set of locations in question are contained in at least l different semantic regions (Xue et al., 2009). Dorfmeister et al. (2015) proposed a method for achieving l-diversity with respect to the source and destination of a route where the client requests a route between regions such that both source and destination satisfy l-diversity. That is, these large regions contain l different semantic regions. The client subsequently locally performs the necessary route computation within these regions. Zhang et al. (2012) proposed a general framework for adjusting the parameters of a given location obfuscation method. In this framework a client specifies the degree of obfuscation required, and this is used in turn to specify access control constraints and adjust the parameters of the method.

As discussed above, a distributed computing paradigm has previously been considered in several methods for online route planning toward achieving robustness to privacy inferences. In these methods computation is distributed across multiple clients. The method proposed in this paper also employs a distributed computing paradigm, but is fundamentally different to the above. Instead of computation being distributed across multiple clients, it is distributed across multiple independent service providers.

3. Distributed location obfuscation method

This section describes the proposed location obfuscation method and is structured as follows. Section 3.1 states all assumptions made with respect to the client and service provider. Section 3.2 describes the method in question.

3.1. Method assumptions

The proposed method makes the assumption that there exists a single client and a set of online route planners. The following assumptions are made with respect to each of these parties.

Client

The client requires a route from a source location s to a destination location d which satisfies the constraints that it is reasonably short and follows the street network. We denote any such route as $(s,\ d)$ where the notation $(.,.)$ denotes a route between two locations satisfying the constraints in question. We assume that both s and d lie within semantic regions. The client does not have any prior knowledge of the street network or the ability to perform route planning in a street network. Instead the client wishes to obtain the route $(s,\ d)$ from the set of online route planners.

The set of online route planners are known to the client to provide a reliable service. However the client has concerns regarding the potential of these service providers to perform inferences with respect to s and d. The client therefore wishes to obtain the route $(s,\ d)$ in a manner which is robust to such inferences. They are willing to accept some reduction in quality of service with respect to route length in order to achieve this robustness. Toward this goal the client decomposes the route $(s,\ d)$ into a sequence of n shorter routes $D = \{(s_1, d_1)\ldots(s_n, d_n)\}$ such that $s_1 = s$, $d_n = d$ and $d_i = s_{i+1}$ for $i = 1 \ldots n-1$. Each of these routes is subsequently requested from an independent online route planner. Upon receipt of the routes in question, the client performs the necessary integration to obtain the route $(s,\ d)$. The above steps are specified precisely in Section 3.2.

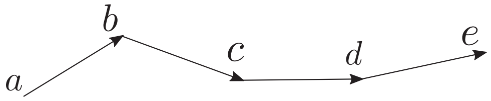

Fig. 2. A route decomposition $D = \{(a, b), (b, c), (c, d), (d, e)\}$ where each individual route is represented by a vector from source to destination.

Online route planner

Each online route planner has an accurate weighted graph model of the street network Bondy and Murty, 1976. This model is denoted by $G = (V, E)$ where V and E are the vertices and edges of the graph respectively. The vertices V correspond to street intersections and deadends, while the edges E correspond to street segments connecting pairs of vertices. This is a commonly used street network model known as a *primary representation* (Corcoran et al., 2015; Hannah et al., 2018). In this work the weight on an edge is equal to the length of the corresponding street segment. However one could also consider a weight equal to travel time. Each online route planner has the ability to perform route planning between any pair of vertices in this model. That is, compute $(.,.)$ for any pair of vertices. This is computed using Dijkstra's algorithm (Mehlhorn and Sanders, 2008). Dijkstra's algorithm is a standard algorithm used for route planning in street networks. However, when dealing with very large networks, it is common to preprocess the network such that the time complexity of subsequent route planning is reduced (Bast et al., 2016). For example *contraction hierarchies* is one such technique which preprocesses the network by adding shortcut edges (Geisberger et al., 2012).

The online route planners are aware that the client is employing the proposed method for online route planning and they wish to infer s and d. If the online route planners do not coordinate and share information, each will have knowledge of at most a single route request in the route decomposition D. On the other hand, a subset of route planners may coordinate and share route requests they receive. In such cases a single online route planner may have knowledge of a subset of route requests in the route decomposition D. Let R denote this subset and m be the number of elements it contains where $1 \leq m \leq n$. Note that, if $m \leq 1$ this corresponds to the case where online route planners do not coordinate and share route requests.

We assume that an online route planner can define a total order on R which is consistent with the total order defined on D.[1] Given this order they can determine the route $(s_s,\ d_s) \in R$ such that $s \leq j$ for all $(s_j,\ d_j) \in R$ and the route $(s_d,\ d_d) \in R$ such that $d \geq j$ for all $(s_j,\ d_j) \in R$. That is, they can determine the routes in R which occur earliest and latest respectively in the route decomposition D. To illustrate this consider the route decomposition $D = \{(a, b), (b, c), (c, d), (d, e)\}$ displayed in Fig. 2, where each individual route is represented by a vector from source to destination. If an online route planner has knowledge of $R = \{(a, b), (b, c), (d, e)\}$ we assume they can determine that $(a,\ b)$ appears before $(b,\ c)$ and

[1] More formally, they can define a monotone map between the total orders on R and D (see Definition 1.59 in Fong and Spivak, 2018)

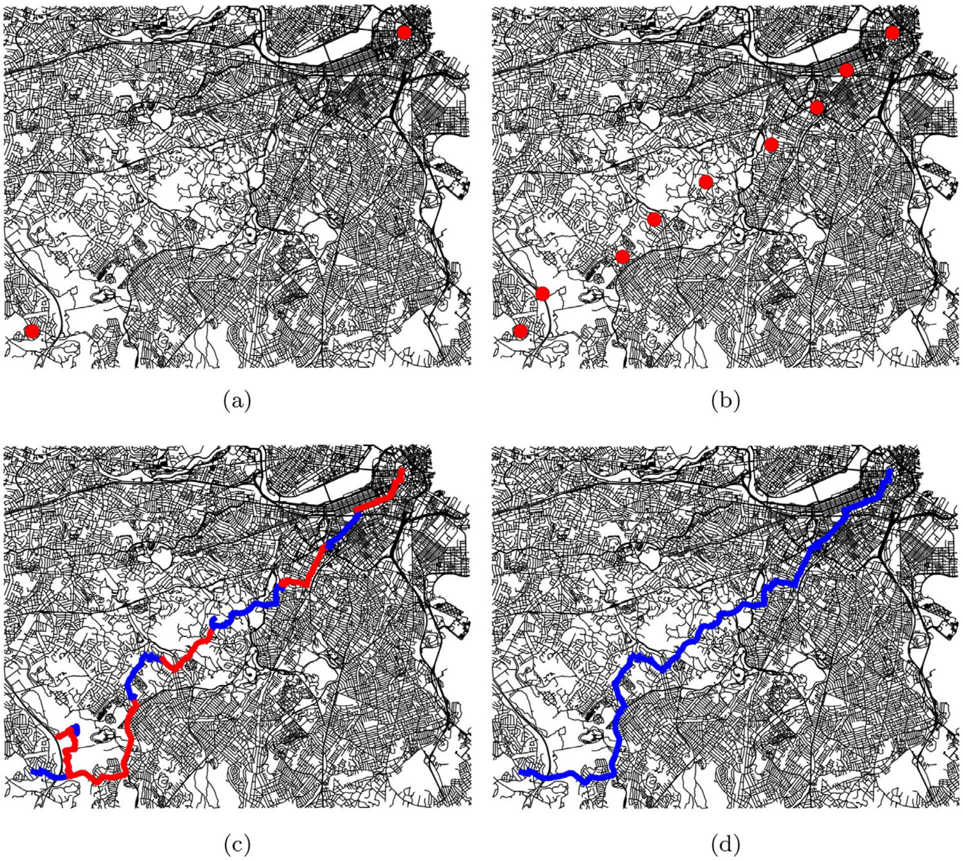

Fig. 3. For the Boston street network in (a), the source location s and destination location d for a route required by a client are represented by red dots in the lower left and upper right respectively. For $n = 8$ the sequence of interpolating points $\{t_i\}_{i=0}^{n}$ between these locations are represented by red dots in (b). The corresponding sequence of routes $D = \{(s_1, d_1) \ldots (s_8, d_8)\}$ are represented in (c) using alternating colors of blue and red. The result of integrating these routes is represented in (d). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

(d, e) in D. Furthermore, we assume they can also determine that (d, e) appears after (a, b) and (b, c) in D.

An online route planner has no knowledge of n, the number of routes that (s, d) has been decomposed into. However, they have sufficient knowledge which allows them to either place a prior estimation or obtain an upper bound on this value. A prior estimate on n could be obtained by surveying a set of individuals who use the service and using the empirical distribution of n as an estimate of the true distribution (Efron and Tibshirani, 1994). An upper bound on n could be obtained by performing an online search for online route planners and counting the number of results retrieved. Let $\mathbb{1}(.)$ denote an indicator function which indicates if a given location lies in a semantic region. The online route planners have the ability to accurately evaluate this function at all source and destination locations in R. We assume the online route planners have no knowledge of previous route requests made by a given client. If the client wishes to request multiple routes, this assumption may be satisfied by making all requests via a trusted anonymizer. Finally, the online route planners are semi-honest meaning that given a client's request they return an accurate result and will not return an incorrect result toward gaining knowledge (Dorfmeister et al., 2015; Xue et al., 2009).

3.2. Method description

The proposed method for online route planning contains three steps. In the first step the client decomposes the route (s, d) into a sequence of n routes $D = \{(s_1, d_1) \ldots (s_n, d_n)\}$ such that $s_1 = s$, $d_n = d$ and $d_i = s_{i+1}$ for $i = 1 \ldots n - 1$. In the second step the client obtains each of these routes from an independent online route planner. In the third step the client integrates this sequence of routes to form a single route (s, d). We now describe in turn how each of these three steps are implemented.

The client decomposes the route (s, d) using the following approach. The client first linearly interpolates the route using a sequence of $n + 1$ locations $\{t_i\}_{i=0}^{n}$ between s and d using Eq. (1).

$$t_i = s + i \times \left(\frac{d - s}{n} \right) \tag{1}$$

Those locations in the subsequence $\{t_i\}_{i=1}^{n-1}$ are subsequently perturbed by the addition of Gaussian noise. The Gaussian in question has mean equal to the zero vector and covariance matrix equal to a diagonal matrix with diagonal elements equal to 500 meters. It is assumed that by adding noise the direction in which the client is traveling is obscured. Finally, the client assigns s_i and d_i to the locations t_{i-1} and t_i respectively for $i = 1 \ldots n$.

Given the above decomposition the client obtains each route in D from an independent online route planner. Here independence is achieved by selecting without replacement n online route planners from the set of all available online route planners.

To illustrate the above two steps consider the Boston street network which is shown in Fig. 3(a). In this figure s and d corresponding to an example route are represented by red dots in the lower left and upper right corners respectively. For $n = 8$ the corresponding set of locations $\{t_i\}_{i=0}^{8}$ are represented by red dots in Fig. 3(b). The corresponding sequence of routes $D = \{(s_1, d_1) \ldots (s_8, d_8)\}$ obtained from independent online route planners are illustrated in Fig. 3(c) where (s_i, d_i) is represented by the color blue if i (mod 2) = 1 and the color red otherwise. Note each of these routes is in fact the shortest route between the source and destination locations in question.

Having obtained the sequence of routes D, the client integrates this sequence to form a single route (s, d). If the client performs this integration in a naive manner by concatenating the routes, the resulting route (s, d) may contain redundant detours. To illustrate this consider the toy street network in Fig. 4(a) and the situation where s and d correspond to the red dots in the upper left and upper right of the figure respectively. For $n = 2$ the corresponding interpolated locations $\{t_i\}_{i=0}^{2}$ are illustrated in Fig. 4(b) using red dots. The routes (s_1, d_1) and (s_2, d_2) between these points are illustrated in Fig. 4(c) and (d) using the colors blue and red respectively. If these routes are integrated by concatenating (s_1, d_1) and (s_2, d_2), the resulting route (s, d) will contain a redundant detour. This is illustrated in Fig. 4(e) where the route in question is represented by the color blue.

To overcome this issue we integrate the sequence of routes using the algorithm described in Algorithm 1 which removes any detours. Let vertex_sequence(.) be a function which maps a route to its corresponding sequence of vertices in the street network graph G. In lines 2 to 5 the algorithm first represents each route (s_i, d_i) by its corresponding sequence of vertices and concatenates these sequences to form a sequence entitled route. In lines 6 to 9 the algorithm next computes a function highest_index(.) using a dictionary data structure which maps each vertex in the sequence route to the index at which it last appears in the sequence. Note that, the statement route(i) returns the vertex at location i in the sequence route. In lines 10 to 20 the algorithm constructs a new sequence entitled route_new by removing all subsequences from the sequence route between the first and last appearance of each vertex. The sequence route_new corresponds to a route without detours and is returned in line 21.

Algorithm 1: Integration of routes.

```
    Input: A sequence of n routes $(s_1, d_1)\dots(s_n, d_n)$ such that
           $s_1 = s$, $d_n = d$ and $d_i = s_{i+1}$ for $i = 1\dots n-1$.
    Output: A single route $(s, d)$ containing no detours such that
            $s_1 = s$ and $d_n = d$.
 1  begin
 2      route = ()
 3      for $i \leftarrow 1$ to $n$ do
 4          route.append(vertex_sequence(($s_i$, $d_i$)))
 5      end

 6      highest_index = dict()
 7      for $i \leftarrow 1$ to size(route) do
 8          highest_index(route($i$)) = $i$
 9      end

10      route_new = ()
11      $i = 1$
12      while $i \le$ size(route) do
13          $v$ = route($i$)
14          route_new.append($v$)
15          if highest_index($v$) > $i$ then
16              $i$ = highest_index(v) + 1
17          else
18              $i = i + 1$
19          end
20      end

21      return route_new
22  end
```

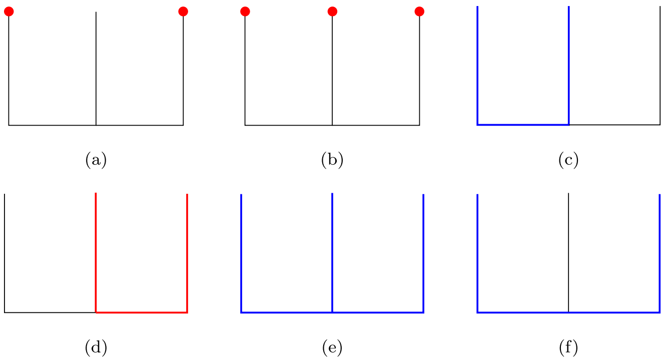

Fig. 4. For the example street network in (a), the source location s and destination location d for a route required by a client are represented by red dots in the upper left and upper right respectively. For $n = 2$ the sequence of interpolated points {t_0, t_1, t_2} between these locations are represented by red dots in (b). The corresponding sequence of routes (s_1, d_1) and (s_2, d_2) are represented in (c) by the color blue and (d) by the color red respectively. The result of integration in a naive manner is illustrated in (e) while the result of integration using the proposed solution is illustrated in (f). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

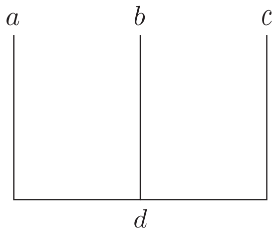

Fig. 5. An illustration of the street network in Fig. 4 where its vertices have been labeled a, b, c and d is displayed.

To illustrate this integration consider again the case of computing a route in the toy street network of Fig. 4. A representation of this street network where its vertices have been labeled a, b, c and d is displayed in Fig. 5. Given this labelling, the representations of (s_1, d_1) and (s_2, d_2) in terms of their corresponding sequence of vertices are [a, d, b] and [b, d, c] respectively. Concatenating these sequences to form a single sequence gives [a, d, b, b, d, c]. The highest indices at which the vertices a, b, c and d appear are 1, 4, 6 and 5 respectively. Removing all subsequences between the first and last appearance of each vertex results in the sequence [a, d, c] which contains no detours. This route is represented by the color blue in Fig. 4(f).

4. Evaluation

This section presents an evaluation of the proposed location obfuscation method for online route planning. Specifically we evaluate the method with respect to robustness to privacy inferences, time and communication complexity, and quality of service. These three aspects are discussed in Sections 4.1, 4.2 and 4.3 respectively. This evaluation includes a number of experiments which were performed using the Boston street network illustrated in Fig. 3(a) and obtained from OpenStreetMap (Corcoran et al., 2013).

4.1. Robustness

In this section we demonstrate that, without significant coordination and sharing of route requests, an online route planner cannot infer s and d with high confidence. Recall from Section 3.1 that an online route planner has knowledge of R which is a subset of the route decomposition D. Furthermore, recall that they can determine $(s_s, d_s) \in R$ and $(s_d, d_d) \in R$ which correspond to the routes which occur earliest and latest respectively in the route decomposition D.

Let D_s and D_d denote the set of source and destination locations respectively in D. Also, let R_s and R_d denote the set of source and destination locations respectively in R. The route (s_s, d_s) is the only route in R which may equal (s_1, d_1) in D. In turn, s_s is the only location in R_s which may equal s. Similarly the route (s_d, d_d) is the only route in R which may equal (s_n, d_n) in D. In turn, d_d is the only location in R_d which may equal d. In what follows we present inferences with respect to s_s. A similar inference applies with respect to d_d.

As stated in Section 3.1, we assume an online route planner has sufficient knowledge which allows them to either place a prior estimation or obtain an upper bound on the value of n, the number of routes that (s, d) has been decomposed into. We first consider the case where the online route planner has a prior estimation of n.

If s_s does not lie in a semantic region, that is $\mathbb{1}(s_s) = 0$, then the online route planner wishes to estimate $P(s_s = s \mid \mathbb{1}(s_s) = 0, R, n)$. Since we assume s lies in a semantic region this implies that $s_s \neq s$. That is, since s lies in a semantic region but s_s does not they cannot be the same location. The probability $P(s_s = s \mid \mathbb{1}(s_s) = 0, R, n)$ is therefore zero and the online route planner cannot infer s given R.

If s_s lies in a semantic region, that is $\mathbb{1}(s_s) = 1$, then the online route planner wishes to estimate $P(s_s = s \mid \mathbb{1}(s_s) = 1, R, n)$. Let T denote the set of elements in D_s which may equal s. Eq. (2) defines the expected size of T where $|.|$ is the set size operator. The factor $n - m$ equals the number of locations in the set $D_s - R_s$. The factor $P(\mathbb{1}(s_i) = 1 \mid s_i \in D_s)$ equals the probability that an element in this set lies in a semantic region and therefore may equal s. The added term 1 corresponds to the fact that $s_s \in D_s$, where $\mathbb{1}(s_s) = 1$ is the single element in $R_s \subset D_s$ which may equal s.

$$\mathbb{E}[|T|] = (n - m)P(\mathbb{1}(s_i) = 1 \mid s_i \in D_s) + 1 \tag{2}$$

Toward estimating the term $P(\mathbb{1}(s_i) = 1 \mid s_i \in D_s)$ in Eq. (2), we perform the factorization in Eq. (3).

$$\begin{aligned} &P(\mathbb{1}(s_i) = 1 \mid s_i \in D_s) \\ &\quad = P(\mathbb{1}(s_i) = 1 \mid s_i \in D_s, s_i = s)P(s_i = s) \\ &\qquad + P(\mathbb{1}(s_i) = 1 \mid s_i \in D_s, s_i \neq s)P(s_i \neq s) \\ &\quad = 1\frac{1}{n} + P(\mathbb{1}(s_i) = 1 \mid s_i \in D_s, s_i \neq s)\frac{n-1}{n} \\ &\quad = \frac{1}{n} + P(\mathbb{1}(s_i) = 1 \mid s_i \in D_s, s_i \neq s)\frac{n-1}{n} \end{aligned} \tag{3}$$

Toward estimating the term $P(\mathbb{1}(s_i) = 1 \mid s_i \in D_s, s_i \neq s)$ in Eq. (3) we note that a location satisfying the conditions $s_i \in D_s$ and $s_i \neq s$ is a random interpolated location. We therefore estimate this probability by sampling a set of random locations in the environment and computing the proportion of which lie in semantic regions. By the *law of large numbers* this estimate converges to the true probability as the number of random locations increases. In order to determine if a given random location lies in a semantic region we used the following approximation which represents a lower bound. We first determine the street which is closest to the location in question. If this street is of the type *residential* then we classify the location as lying in a semantic region. Otherwise we classify the location as not lying in a semantic region. This approximation represents a lower bound because it only considers streets and ignores other features of the environment which would result in a point being classified as lying in a semantic region. For example if the point in question lies in the grounds of a hospital it should be classified as lying in a semantic region irrespective of the type of street it is closest to. Applying the above approximation in the context of the Boston street network and using 10,000 random locations, the probability $P(\mathbb{1}(s_i) = 1 \mid s_i \in D_s, s_i \neq s)$ was estimated to be 0.36.

Table 1
The probability values $P(s_s = s \mid \mathbb{1}(s_s) = 1, R, n)$ and $P(d_d = d \mid \mathbb{1}(d_d) = 1, R, n)$ for different values of n and m are displayed. Here n is the number of routes in the decomposition D, R is the subset of D which the online route planner has knowledge of and m is the number of routes in R.

	$n = 1$	$n = 2$	$n = 4$	$n = 8$	$n = 16$	$n = 32$	$n = 64$
$m = 1$	1.00	0.59	0.39	0.24	0.14	0.07	0.04
$m = 2$		1.00	0.49	0.27	0.15	0.08	0.04
$m = 4$			1.00	0.36	0.17	0.08	0.04
$m = 8$				1.00	0.23	0.09	0.04
$m = 16$					1.00	0.14	0.05
$m = 32$						1.00	0.07
$m = 64$							1.00

Substituting the above results into Eq. (2) gives us the estimated expectation in Eq. (4).

$$\mathbb{E}[|T|] = (n - m)\left(\frac{1}{n} + 0.36\frac{n-1}{n}\right) + 1 \tag{4}$$

If we assume each element in the set T equals s with equal probability, the probability $P(s_s = s \mid \mathbb{1}(s_s) = 1, R, n)$ is defined in Eq. (5).

$$P(s_s = s \mid \mathbb{1}(s_s) = 1, R, n) = \frac{1}{\mathbb{E}[|T|]} \tag{5}$$

To illustrate the above inferences consider an example where $n = 10$ and $m = 10$. That is, the online route planner has knowledge of all route requests in the route decomposition $D = \{(s_1, d_1) \ldots (s_n, d_n)\}$. The online route planner can determine s_s and if $\mathbb{1}(s_s) = 1$ they can in turn infer that $P(s_s = s \mid \mathbb{1}(s_s) = 1, R, n) = 1$. A similar inference can be applied to the location d. Consider a second example where $n = 32$ and $m = 1$. That is, the online route planner has knowledge of a single route request. The online route planner can determine s_s and if $\mathbb{1}(s_s) = 1$ they can in turn infer that $P(s_s = s \mid \mathbb{1}(s_s) = 1, R, n) = 0.07$. Again a similar inference can be applied to the location d. The probability values $P(s_s = s \mid \mathbb{1}(s_s) = 1, R, n)$ and $P(d_d = d \mid \mathbb{1}(d_d) = 1, R, n)$ for different values of n and m are displayed in Table 1. From these probability values we note that coordinating online route planners cannot infer s or d with high confidence provided that m is relatively smaller than n.

In the above inferences we assume the online route planner has a prior estimation on the value of n. Let us assume now that this is not the case and instead the online route planner has only an upper bound u on this value. In this case the online route planner may assign a uniform distribution over the possible values of n=m,...,u. That is, $P(n) = 1/(u - m + 1)$. Note that, m is a lower bound for n because this is the number of route requests the online route planner has knowledge of. Given $\mathbb{1}(s_s) = 1$, the online route planner may attempt to estimate $P(s_s = s \mid \mathbb{1}(s_s) = 1, R, u)$ by performing a marginalization of n using Eq. (6). Again, an identical inference applies with respect to d_d.

$$P(s_s = s \mid \mathbb{1}(s_s) = 1, R, u) = \sum_{n=m}^{u} P(s_s = s \mid \mathbb{1}(s_s) = 1, R, n)P(n) \tag{6}$$

The probability values $P(s_s = s \mid \mathbb{1}(s_s) = 1, R, u)$ and $P(d_d = d \mid \mathbb{1}(d_d) = 1, R, u)$ for different values of u and m are displayed in Table 2. From these probability values, we note that

Table 2
The probability values $P(s_s = s \mid \mathbb{1}(s_s) = 1, R, u)$ and $P(d_d = d \mid \mathbb{1}(d_d) = 1, R, u)$ for different values of u and m are displayed. Here u is an upper bound on n the number of routes in the decomposition D, R is the subset of D which the online route planner has knowledge of and m is the number of routes in R.

	$u = 32$	$u = 64$	$u = 128$	$u = 256$	$u = 512$	$u = 1024$
$m = 1$	0.20	0.13	0.08	0.04	0.02	0.01
$m = 2$	0.21	0.13	0.08	0.04	0.02	0.01
$m = 4$	0.23	0.14	0.08	0.04	0.02	0.01
$m = 8$	0.25	0.15	0.08	0.05	0.02	0.01
$m = 16$	0.33	0.16	0.09	0.05	0.02	0.01
$m = 32$	1.00	0.22	0.10	0.05	0.03	0.01

coordinating online route planners cannot infer s or d with high confidence provided that m is relatively smaller than u.

4.2. Computational and communication complexity

In this section we evaluate the computational and communication complexity of the proposed method with respect to the online route planners and the client.

We first consider the computational complexity of the method with respect to the online route planners. We prove that the computational complexity of running Dijkstra's algorithm n times to compute the routes (s_i, d_i) is proportional to running Dijkstra's algorithm a single time to compute the route (s, d). That is, the proposed method for computing routes in a distributed manner has computational complexity proportional to traditional models which compute routes in an undistributed manner. Let $\|(x, y)\|$ and $\|x - y\|$ denote the length of the shortest route and Euclidean distance respectively between locations x and y. To simplify our analysis we make the reasonable assumption that for any pair of locations x and y, $\|(x, y)\|$ is proportional to $\|x - y\|$. Let (s_i, d_i) be a route request made by a client to an online route planner. As discussed in Section 3.1, the online route planner computes the route in question using Dijkstra's algorithm where the weights on the graph edges correspond to street length. A property of Dijkstra's algorithm is that the computational complexity of computing (s_i, d_i) is proportional to $\|(s_i, d_i)\|$ (Lee et al., 2009; Mehlhorn and Sanders, 2008). This is due to the fact that the algorithm searches the space of shortest paths in a manner which considers shorter paths first.

The fact that $\|(s_i, d_i)\|$ is proportional to $\|s_i - d_i\|$ is stated in Eq. (7a) and this implies Eq. (7b). Since linear interpolation of points between s and d was employed, the sum of $\|s_i - d_i\|$ is equal to $\|s - d\|$. This implies the result stated in Eq. (7c). Since $\|s - d\|$ is proportional to $\|(s, d)\|$, this implies Eq. (7d). Given that the computational complexity of computing a route is proportional the length of that route, Eq. (7d) implies that in the case of Dijkstra's algorithm the computational complexity of computing the sequence of routes (s_i, d_i) is proportional to that of computing the single route (s, d). The computational complexity of computing the route (s, d) using Dijkstra's algorithm is $O(|E| + |V|log|V|)$ where $|V|$ and $|E|$ are the number of vertices and edges in the street network graph G (Lee et al., 2009; Mehlhorn and Sanders, 2008).

$$\|(s_i, d_i)\| \propto \|s_i - d_i\| \tag{7a}$$

$$\sum_{i=1}^{n} \|(s_i, d_i)\| \propto \sum_{i=1}^{n} \|s_i - d_i\| \tag{7b}$$

$$\sum_{i=1}^{n} \|(s_i, d_i)\| \propto \|s - d\| \tag{7c}$$

$$\sum_{i=1}^{n} \|(s_i, d_i)\| \propto \|(s, d)\| \tag{7d}$$

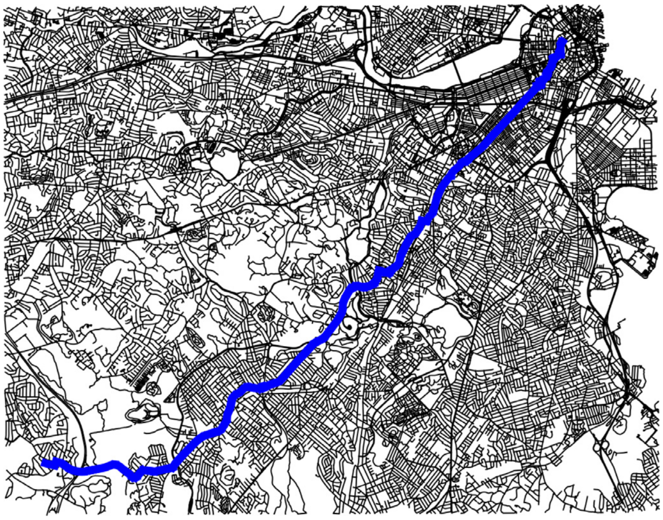

Fig. 6. The shortest route from s to d is displayed.

Table 3
Mean and standard deviation of 1000 route lengths measured in meters computed using the proposed method for different parameter values of n.

	Mean length	Std length
Distributed $n = 1$	9386	3821
Distributed $n = 2$	9685	3889
Distributed $n = 4$	10,161	4010
Distributed $n = 8$	10,789	4352
Distributed $n = 16$	11,646	4697

We next consider the computational complexity of the model with respect to the client. Having received the sequence of n routes (s_i, d_i) from independent online route planners, the client performs integration to form a route from s to d. The algorithm described in Section 3.2 for performing this integration performs two iterations over the sequence of vertices in the concatenation of the n routes $\{(s_1, d_1)\dots(s_n, d_n)\}$. The computational complexity of this integration is therefore $O(m)$ where m is the number of vertices in the sequence. Finally the complexity of communication between the client and online route planners is also $O(m)$.

4.3. Quality of service

The routes computed using the proposed location obfuscation method will in many cases be different from the shortest route from s to d. For example, consider again the example illustrated in Fig. 3(a). Fig. 3(d) displays the corresponding route computed using the proposed method with the parameter $n = 8$. On the other hand, Fig. 6 displays the corresponding shortest route; that is, the route computed using the proposed method with the parameter $n = 1$. It is evident that these two routes are different, and in fact the route computed using the proposed method with parameter $n = 8$ is quite different from the shortest route. This discrepancy can be attributed to the fact that the proposed method integrates a sequence of locally shortest routes and therefore may not return the globally shortest route. As such, the proposed method offers a lower quality of service relative to traditional undistributed models. Here quality of service is measured as the difference between the length of the route returned and the length of the shortest route.

In order to quantify the quality of service of the proposed method we randomly sampled 1000 pairs of s and d locations. Table 3 displays the mean and standard deviation of the corresponding route lengths measured in meters where the routes were computed by the proposed method using different values of the parameter n. Note that, a route computed using the proposed method with parameter $n = 1$ corresponds to the shortest route from s to d. From this table we see that as the parameter n increases the mean and standard deviation of the routes also increases. However the order of this increase is not significant. For example, the mean length of the routes computed by the proposed method with parameter values $n = 1$ and $n = 16$ was 9386 m and 11646 m respectively, i.e. approximately a 25% increase. This represents a relatively small reduction in quality of service in return for increased robustness to privacy inferences.

5. Conclusions

This paper proposes a novel location obfuscation method for online route planning which employs a distributed computing paradigm to achieve robustness to inferences regarding route source and destination. Although the use of such a paradigm has previously been considered, the method proposed in this paper is fundamentally different from existing methods. Instead of computation being distributed across multiple clients, as is the case in existing methods, it is distributed across multiple independent service providers.

It is important to note that the proposed location obfuscation method is not applicable in all situations. It naturally requires that the client has access to a set of independent online route planners, who do not perform significant coordination and sharing of information. In some situations, this requirement may not be satisfied, and the use of an alternative location obfuscation method should be considered. In this and other cases, the proposed method could also be integrated with other location obfuscation methods, such as submitting additional fake route requests to the service providers, to further increase robustness to privacy inferences.

Given the original nature of this work, there exists many potential avenues for future research. One avenue would be the development of more sophisticated means for decomposing the required route into a sequence of shorter route requests. In the proposed method a very simple linear interpolation with the addition of noise was employed. However, robustness to privacy inferences could be improved if prior knowledge regarding the locations of semantic regions was used to ensure interpolated points lie in such regions. Such prior knowledge could be obtained by querying an additional service provider. Another avenue for future research would be the development of a method which, as mentioned above, integrates the proposed method with other location obfuscation methods.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cose.2020.101850.

References

Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., Werneck, R.F., 2016. Route Planning in Transportation Networks. In: Algorithm Engineering. Springer, pp. 19–80.

Bondy, J.A., Murty, U.S.R., 1976. Graph Theory with Applications, 290. Macmillan London.

Buchanan, W.J., Kwecka, Z., Ekonomou, E., 2013. A privacy preserving method using privacy enhancing techniques for location based services. Mob. Netw. Appl. 18 (5), 728–737.

Chow, C.-Y., Mokbel, M.F., Bao, J., Liu, X., 2011. Query-aware location anonymization for road networks. Geoinformatica 15 (3), 571–607.

Chow, C.-Y., Mokbel, M.F., Liu, X., 2006. A peer-to-peer spatial cloaking algorithm for anonymous location-based service. In: ACM International Symposium on Advances in Geographic Information Systems, pp. 171–178.

Corcoran, P., Jilani, M., Mooney, P., Bertolotto, M., 2015. Inferring semantics from geometry: the case of street networks. In: SIGSPATIAL International Conference on Advances in Geographic Information Systems, p. 42.

Corcoran, P., Mooney, P., Bertolotto, M., 2013. Analysing the growth of OpenStreetMap networks. Spat. Stat. 3, 21–32.

Dorfmeister, F., Wiesner, K., Schuster, M., Maier, M., 2015. Preventing restricted space inference in online route planning services. In: International conference on mobile and ubiquitous systems: computing, networking and services, pp. 209–218. Brussels, Belgium

Efron, B., Tibshirani, R.J., 1994. An introduction to the bootstrap. CRC press.

Fong, B., Spivak, D. I., 2018. Seven sketches in compositionality: an invitation to applied category theory. preprint arXiv:1803.05316.

Geisberger, R., Sanders, P., Schultes, D., Vetter, C., 2012. Exact routing in large road networks using contraction hierarchies. Transp. Sci. 46 (3), 388–404.

Ghinita, G., Kalnis, P., Skiadopoulos, S., 2007. Mobihide: a mobile peer-to-peer system for anonymous location-based queries. In: International Symposium on Spatial and Temporal Databases. Springer, pp. 221–238.

Ghinita, G., Kalnis, P., Skiadopoulos, S., 2007. Prive: anonymous location-based queries in distributed mobile systems. International Conference on World Wide Web, pp. 371–380.

Gruteser, M., Grunwald, D., 2003. Anonymous usage of location-based services through spatial and temporal cloaking. In: International Conference on Mobile Systems, Applications and Services, pp. 31–42.

Hannah, C., Spasić, I., Corcoran, P., 2018. A computational model of pedestrian road safety: the long way round is the safe way home. Accident Anal. Prevent. 121, 347–357.

Krumm, J., 2007. Inference attacks on location tracks. In: International Conference on Pervasive Computing. Springer, pp. 127–143.

Lee, K.C., Lee, W.-C., Leong, H.V., Zheng, B., 2009. Navigational path privacy protection. In: ACM conference on Information and Knowledge Management, pp. 691–700.

Luo, J.-N., Yang, M.-H., 2017. Unchained cellular obfuscation areas for location privacy in continuous location-based service queries. Wirel. Commun. Mob. Comput. 2017.

Mehlhorn, K., Sanders, P., 2008. Algorithms and Data Structures: the Basic Toolbox. Springer Science & Business Media.

Mooney, P., Corcoran, P., 2012. Using OSM for LBS – an analysis of changes to attributes of spatial objects. Advances in Location-Based Services. Springer, pp. 165–179.

Mouratidis, K., Yiu, M.L., 2010. Anonymous query processing in road networks. IEEE Trans. Knowl. Data Eng. 22 (1), 2–15.

Mouratidis, K., Yiu, M.L., 2012. Shortest path computation with no information leakage. VLDB Endowment 5 (8), 692–703.

Rodden, T., Friday, A., Muller, H., Dix, A., et al., 2002. A lightweight approach to managing privacy in location-based services. Technical Report. Equator-02-058. CSTR-07-006.

Shokri, R., Theodorakopoulos, G., Papadimitratos, P., Kazemi, E., Hubaux, J.-P., 2014. Hiding in the mobile crowd: location privacy through collaboration. IEEE Trans. Depend. Secure Comput. 11 (3), 266–279.

Vicente, C.R., Assent, I., Jensen, C.S., 2011. Effective privacy-preserving online route planning. In: IEEE International Conference on Mobile Data Management, 1, pp. 119–128.

Xi, Y., Schwiebert, L., Shi, W., 2014. Privacy preserving shortest path routing with an application to navigation. Pervasive Mob. Comput. 13, 142–149.

Xu, C., Zhu, L., Liu, Y., Guan, J., Yu, S., 2018. Dp-ltod: differential privacy latent trajectory community discovering services over location-based social networks. IEEE Trans. Serv. Comput.

Xue, M., Kalnis, P., Pung, H.K., 2009. Location diversity: Enhanced privacy protection in location based services. International Symposium on Location- and Context-Awareness. Springer, pp. 70–87.

Zhang, Y., Chen, K., Lian, Y., 2012. A path-based access control method for location obfuscation in mobile environment. In: 2012 IEEE Symposium on Electrical & Electronics Engineering, pp. 570–573.

Zhong, G., Hengartner, U., 2008. Toward a distributed k-anonymity protocol for location privacy. In: ACM Workshop on Privacy in the Electronic Society, pp. 33–38.

Dr Padraig Corcoran is a Senior Lecturer in the School of Computer Science and Informatics at Cardiff University. His main research interests are in the fields of graph theory, applied topology and operations research. He is particularly interested in applications to the domains of geographical data science and transportation.

Dr. Peter Mooney is a Lecturer at the Department of Computer Science in Maynooth University, Ireland. His main research interests focus on the analysis and usage of geospatial data from sources such as social media, Volunteered Geographic Information (VGI), and citizen science using techniques such as database

technologies, machine learning and deep learning. In recent years he has led a number of European initiatives which considered the quality and application of VGI in society.

Dr. Andrei Gagarin is a Lecturer in Mathematics at Cardiff University. He received his PhD in Computer Science from the University of Manitoba in 2003, after completing MSc degrees in Mathematics at Belarusian State University (Minsk) and in Operational Research, Combinatorics, and Optimisation at the National Polytechnic Institute of Grenoble (INPG) / Joseph Fourier University. In 2003-06, he was a post-doctoral research fellow in Combinatorics and Bioinformatics at the University of Quebec in Montreal. In 2006-13, he worked in the Department of Mathematics and Statistics and the School of Computer Science of Acadia University. In 2013-16, he was a Research Assistant at Royal Holloway, University of London. His main research interests are in graph theory, optimisation in networks, combinatorics, operational research, data analysis, algorithms design and engineering, workflows and access control.