On the RSU-based Secure Distinguishability Among Vehicular Flows

Wei Chang*, Huanyang Zheng**, and Jie Wu**
* Department of Computer Science, Saint Joseph's University, Philadelphia, PA 19131
** Department of Computer and Information Sciences, Temple University, Philadelphia, PA 19122
Email: wchang@sju.edu, {huanyang.zheng, jiewu}@temple.edu

***Abstract*—Within future smart cities, people are expecting the usage of data from moving vehicles. Due to the existence of malicious users, who claim to be in a traffic flow but actually were in others, a location proof for vehicular trajectory-based data is needed. RoadSide Units (RSUs) are commonly used in smart cities, and a vehicular trajectory's location proof can be generated based on messages collected from RSUs along the trajectory. This paper studies the optimal RSU placement problem: Given a set of traffic flows, the objective is to place a minimum number of RSUs to securely distinguish all of them. A traffic flow is securely distinguishable if the set of its passing RSUs is unique among all traffic flows and unforgeable from each other. To solve the problem, an RSU placement algorithm with an approximation ratio $O(\ln n)$ is proposed. In order to further reduce the number of deployed RSUs, this paper explores the credential propagation mechanism via Car-to-Car (C2C) communications, which essentially extend the coverage of an RSU. Approximation algorithms are proposed to solve the problem, and extensive real data-driven experiments demonstrate the efficiency and effectiveness of the proposed algorithms.**

***Index Terms*—Location proof, optimal placement, priority level, roadside unit, vehicular trajectory.**

I. Introduction

Within future *Smart City*, people are expecting the usage of data not only from static roadside sensors but also from vehicles moving within the cities. Unlike the conventional Location-Based Services (LBS), where data is related to a single location spot, the vehicular trajectory-based data is a continuous observation along the vehicle's trajectory. In a foreseeable future, this vehicular trajectory-based data will support a considerable number of new applications, such as criminal scene reconstruction, smart traffic flow monitoring, and environmental monitoring.

However, using trajectory-based data also brings new security issues. From the existing LBS and mobile social networks, we have already seen the motivations for an adversary to misstate their location claims [1], [2]. Consequently, a key requirement for the next generation of the smart city is the capability to verify trajectory-based location claims made by vehicles. The problem scenario of this paper is illustrated by Fig. 1. There are three predefined traffic flows, f_1 to f_3. A malicious user, who was driving along f_1, tries to pretend that he was in f_2 or f_3. A mechanism is needed to against such a false claim about the moving trajectory.

A RoadSide Unit (RSU) is a typical infrastructure widely adopted in smart cities. Depending on the applications, an RSU may send messages to passing vehicles via wireless networks, or monitor traffic flows by sensors and cameras. In this paper, we consider the scenarios where RSUs are used to develop a "location proof" for vehicles. The location proof is a means for a vehicle to demonstrate that it was indeed in a specific traffic flow. RSUs are deployed on a certain road stretch (i.e. street) and broadcast their unique RSU IDs via RSU-messages to passing vehicles. For instance, in Fig. 1, the RSU on e_4 sends an RSU-message to a passing car, and the car extracts the corresponding RSU ID from the message. The location proof for a vehicular trajectory is created based on the collected RSU IDs along its moving path. When a vehicle claims to be in a specific vehicle flow, we would be able to verify the claim by comparing its collected RSU IDs against a known database of every RSU's geographic information. Malicious users, who were in other traffic flows but not in the claimed one, should be unable to obtain/generate the correct set of RSU IDs.

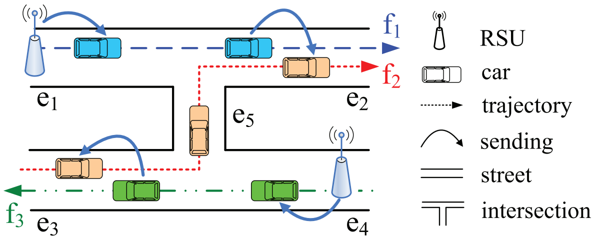


Fig. 1. Securely distinguish three vehicle flows.

Ideally, we could place RSUs on every road stretch that is passed by given traffic flows. However, the manufacturing cost of such an RSU placement strategy is too high since RSUs are expensive. We should minimize the number of placed RSUs to reduce the cost, and in the meantime, different traffic flows can still be *securely distinguished* from each other. Here, the secure distinguishability means that a malicious user cannot pretend to be in other traffic flows by using the RSU IDs received along his actual movement trajectory.

In this paper, we find that, in order to provide the secure distinguishability among given flows, the deployed RSUs should not only be able to cover and uniquely distinguish every flow but also guarantee that the set of passing RSUs for any flow is not a subset of any other flows. The coverage, distinguishability, and non-subset requirements pose unique challenges on our problem. We show that the optimal RSU placement problem is NP-hard and monotonic, but not

978-1-5386-2704-4/17/$31.00 © 2017 IEEE

submodular. It is a non-trivial extension of the traditional hitting set problem, which is submodular. An approximation algorithm is proposed with an $O(\ln n)$ approximation ratio, where n is the number of flows.

In order to further reduce the number of deployed RSUs, we introduce a new concept, priority level, which reflects the importance of a flow. We explore the credential propagation mechanism via Car-to-Car (C2C) communications, and the secure distinguishability among different prioritized flows is established by the RSU IDs via different hops of propagations. C2C communications essentially extend the coverage of an RSU. Take Fig. 1 as an example. Originally, only the vehicles in flow f_3 can obtain RSU-messages from the RSU on street e_4. By allowing C2C communication, vehicles in flow f_2 can also obtain this RSU's ID. Since the secure distinguishability is achieved based on the uniqueness of an unforgeable set of collected RSU IDs, using a credential propagation mechanism makes such a uniqueness be fulfilled by a smaller set of RSUs. However, in practice, an RSU-message may be lost during the multi-hop relay, and the priority level of the flows may affect the optimal number of RSUs. In this paper, we propose another two algorithms to solve these problems.

II. Basic Model: Optimal RSU Placement

Placing RSUs on every road stretch can provide vehicular location proofs, but it is too expensive in terms of the number of deployed RSUs. For minimizing the deployment costs, an optimal RSU placing algorithm is needed. This section studies a basic model when all vehicular trajectories are treated equally, and the next section studies an advanced model where flows may have different priority levels.

A. Security Requirements for Placing RSUs

The goal of our paper is to use RSU generated location proofs against attackers, who claimed that they had gone along certain paths but actually did not. We assume that there are no colluded attackers, and attackers are not able to forge the location proof generated by an RSU without physically appearing at the RSU's covered area. However, since an attacker is still able to hide/drop certain received RSU-messages in order to pretend that he was somewhere else, the security requirements of the RSU-based location proof for vehicular trajectory data are not trivial.

Take Fig. 2 as an example. Assume that there is a map consisting of eight road stretches (i.e. streets), and there are six vehicular flows, which are given in Table I. Our objective is to install a minimal number of RSUs on certain streets such that every flow can be uniquely identified according to the RSU identity number within the received RSU-messages. For the ease of description, we name the received RSU identities as *tags*. For instance, if an RSU is placed on street e_7 and a flow goes through the street, then every vehicle in the flow is able to obtain an RSU tag e_7.

TABLE I
The received tags of flows in Fig. 2

ID	six given vehicle flows	S_1	S_2	S_3
f_1	$e_1 \to e_7 \to e_5 \to e_6$	$\emptyset$	e_7	e_6, e_7
f_2	$e_4 \to e_5 \to e_6$	e_4	e_4	e_4, e_6
f_3	$e_4 \to e_5 \to e_8 \to e_3$	e_3, e_4	e_4, e_8	e_4, e_8
f_4	$e_1 \to e_2 \to e_8 \to e_6$	e_2	e_8	e_6, e_8
f_5	$e_1 \to e_7 \to e_5 \to e_8 \to e_3$	e_3	e_7, e_8	e_7, e_8
f_6	$e_4 \to e_7 \to e_2 \to e_3$	e_2, e_3	e_4, e_7	e_4, e_7

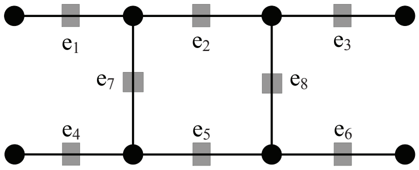


Fig. 2. Securely distinguish six vehicle flows in Table. I. The black vertices represent intersections, edges indicate streets, and the gray boxes give the potential places where an RSU can be deployed.

Table I gives three different RSU placements on Fig. 2. If only honest users are considered, the optimal RSU placement strategy is $\{e_2, e_3, e_4\}$, and the received tags of each flow are given in the "S_1" column of Table I. Clearly, all of them are different, and therefore, the given traffic flows are fully distinguishable. However, this strategy has a problem when the system contains malicious users: any attacker can easily pretend to be flow f_1 by using an empty tag set. Thus, all flows must be covered by at least one tag. Column "S_2" shows another strategy by deploying RSUs on streets e_4, e_7, and e_8, which provides *full distinguishability* and *full coverage* on the given flows. However, in terms of security, these two requirements are still not enough. For the attackers who travel along the flow f_6, they are able to disguise themselves as either f_1 or f_2 by dropping tags from e_4 or e_7, since the received tag set of f_6 is a superset of that of f_1 and f_2. The secure and optimal RSU placement in Fig. 2 deploys RSUs on $\{e_4, e_6, e_7, e_8\}$, and the corresponding tag set of each flow can be found under the "S_3" column. In summary, the optimal placement of the RSUs must guarantee three requirements: full distinguishability, full coverage, and non-subset relations.

B. Problem Formulation for Basic Model

The RSU placement scenario is based on an undirected graph (i.e. map) $G = (V, E)$, where node set V is a set of road intersections, and an undirected edge set $E = \{e\} \subseteq V^2$ represents road stretches on G with $E \subseteq V^2$. We use e_i to denote the ith edge. G contains m predefined vehicle flows $F = \{f_1, f_2, \ldots, f_m\}$. Each flow is represented as a set of streets visited by the flow, $f_i = \{e_1, e_2, \ldots\}$. Note that all flows in F satisfy: $f_i \nsubseteq f_j$ for $\forall f_i, f_j \in F, f_i \neq f_j$.

In order to securely distinguish vehicles from different flows, several RSUs are deployed on E: whenever a vehicle passes an RSU, the vehicle will receive an RSU-message, which contains the unique ID of the RSU [3]. Let S denote an RSU placement strategy, which is the variable in our problem. For instance, in the example of Table I and Fig. 2, the RSU placement strategy is $S_{opt} = \{e_4, e_6, e_7, e_8\}$. Let $S(f)$ denote a set of tags that flow f has received from RSUs under the RSU placement strategy S. So, in Table I,

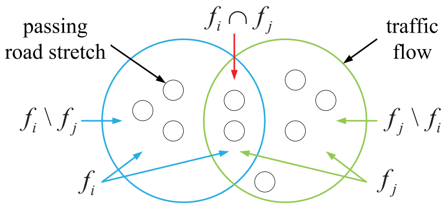
$S_{opt}(f_1) = \{e_6, e_7\}$ and $S_{opt}(f_2) = \{e_4, e_6\}$. We say that flows f_i and f_j are *securely distinguishable* if their received tag sets are not the subset of each other: $S(f_i) \nsubseteq S(f_j)$ and $S(f_j) \nsubseteq S(f_i)$. Note that the strategy that satisfies the secure distinguishability requirement definitely provides full distinguishability (i.e. $S(f_i) \neq S(f_j)$) and full coverage (i.e. $S(f_i) \neq \emptyset, S(f_j) \neq \emptyset$) to the given flows. Considering that RSUs are generally expensive, our objective is to securely distinguish all flows in F by placing a minimum number of RSUs on E. Let $|\cdot|$ denote set cardinality. The optimal RSU placement problem is formulated as follows:

$$\min \quad |S| \tag{1}$$

$$\text{s.t.} \quad S(f_i) \nsubseteq S(f_j) \text{ for } \forall f_i, f_j \in F, f_i \neq f_j \tag{2}$$

C. Problem Analysis

We can divide the set of $f_i \cup f_j$ into three disjoint subsets of $f_i \backslash f_j$, $f_j \backslash f_i$, and $f_i \cap f_j$. Since $f_i \nsubseteq f_j$, we have $f_i \backslash f_j \neq \emptyset$ and $f_j \backslash f_i \neq \emptyset$. These subsets are depicted in the following:



The key observation is formally presented in the following:

***Theorem** 1:* To cover and distinguish an arbitrary pair of traffic flows (f_i and f_j), two RSUs should be placed on streets from two subsets of $f_i \backslash f_j$ and $f_j \backslash f_i$, respectively.

Due to page limitation, we skip all theorems' proof details, which can be found from here [4]. The RSUs, which are not placed on road stretches in $f_i \cup f_j$, will neither cover nor distinguish f_i and f_j. Take flows f_1 and f_2 from Table I as an example, where we have: $f_1 \backslash f_2 = \{e_1, e_7\}$, $f_2 \backslash f_1 = \{e_4\}$, and $f_1 \cap f_2 = \{e_5, e_6\}$. To securely distinguish only f_1 and f_2, we can have $S = \{e_1, e_4\}$, or $S = \{e_4, e_7\}$.

***Theorem** 2:* The optimal RSU placement is NP-hard, monotonic but non-submodular.

D. Approximation for the Optimal RSU Placement

This subsection presents a greedy approximation algorithm. Based on Theorem 1, two RSUs should be placed on streets from two subsets of $f_i \backslash f_j$ and $f_j \backslash f_i$, respectively. Note that we have $f_i \backslash f_j \neq \emptyset$ and $f_j \backslash f_i \neq \emptyset$, since $f_i \nsubseteq f_j$ and $f_j \nsubseteq f_i$. Algorithm 1 is proposed. After the initialization (line 1), it decomposes each pair of traffic flows into two subsets (lines 2 and 3). These subsets are added to D. A counter is maintained for each street (lines 4 and 5). Algorithm 1 iteratively updates an RSU to the current S though a greedy placement (lines 6 to 12). The iteration terminates when all pairs of given traffic flows are covered and distinguishable ($D \neq \emptyset$ in line 6). In each iteration, Algorithm 1 calculates C_e for each street (lines 7 to 9). C_e represents the number of included subsets in D, if an RSU is placed on the street of e. An RSU is placed on a street from each of the two subsets of each traffic flow pair. However, a street, e, may include multiple subsets from different traffic flow pairs. The street, which maximizes C_e, is greedily added to S (line 10). The corresponding subsets in D are removed (line 11). Algorithm 1 resets $C_e = 0$ for the next iteration (line 12). Finally, S is returned (line 13). Algorithm 1 is bounded:

***Theorem** 3:* Algorithm 1 achieves a ratio of $O(\ln n)$ to the optimal algorithm for the number of placed RSUs.

Algorithm 1 Subset-Based Greedy (SBG)

```
1:  Initialize $S = \emptyset$ and $D = \emptyset$.
2:  for each pair of traffic flows, $f_i$ and $f_j$ do
3:      Add $d_{ij} = f_i \backslash f_j$ and $d_{ji} = f_j \backslash f_i$ to $D$.
4:  for each street, $e \in E$ do
5:      Initialize a counter of $C_e = 0$.
6:  while $D \neq \emptyset$ do
7:      for each subset, $d_{ij} \in D$ do
8:          for $e \in d_{ij}$ and $e \in E \backslash S$ do
9:              Update $C_e = C_e + 1$.
10:     Update $S = S \cup \{\arg\max_e C_e\}$.
11:     Remove $d_{ij}$ for $\arg\max_e C_e$ from $D$.
12:     Reset $C_e = 0$ for each street, $e$.
13: return $S$ as the RSU placement strategy.
```

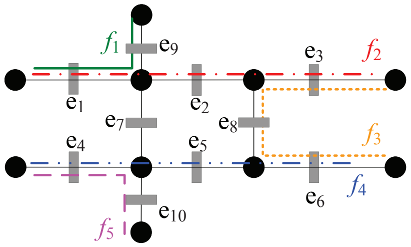


Fig. 3. Distinguish vehicle flows by RSU and C2C.

III. Priority Level-based Optimal RSU Placement

In practice, traffic flows have different densities and importance, and many applications do not need a uniform priority or security requirement for all flows.

A. Priority Level and Car-to-Car Propagation-based Tag

In order to further reduce the number of deployed RSUs while maintaining the secure distinguishability for given vehicular flows, in this section, we propose a new concept, *flow priority level*. We cut off some RSUs that mainly served low priority flows by exploring the Car-to-Car (C2C) communication mechanism. We let the location proofs generated by the RSUs further propagate to other nearby vehicular flows, which do not pass the RSUs. Essentially, we use the credential propagation mechanism to increase an RSU's coverage.

Let l_i denote the priority level of a flow f_i, where $l_i = 0, 1, \ldots, l_{\max}$. The lower the value of l is, the higher priority the flow has. l controls the maximum hops that an RSU's impact can contribute to the corresponding flow's secure distinguishability. From the consideration of the computing complexity, we set $l_{\max}$ as a small integer, such as 1 or 2.

We add a new dimension, *propagation hop*, to the received RSU tags. We use $e_i^{[j]}$ to represent a j-hop propagated credential from the RSU on edge e_i. For a vehicle, which directly passed an RSU on street e_i, the vehicle will possess tag $e_i^{[0]}$. Whenever a vehicle received a tag $e_i^{[j]}$, the vehicle immediately creates a new message, which contains a new tag $e_i^{[j+1]}$, and keeps broadcasting the message to all passing vehicles. Here, we assume the tag propagation process is secure. The propagation terminates when the hop counter reaches $l_{\max}$. For instance, if an RSU is placed at street e_9 of Fig. 3 and $l_{\max} = 1$, vehicles in flow f_1 will obtain $e_9^{[0]}$ directly from the RSU and $e_9^{[1]}$ from other vehicles in the same flow. Since f_1 and f_2 share a common street e_1, vehicles in f_2 will also get $e_9^{[1]}$ from the vehicles in f_1 when they pass each other on e_2. However, flows f_3 to f_5 will not have any tags related to e_9 since the maximum propagation hop is $l_{\max} = 1$.

The C2C-based RSU tags can also provide the secure distinguishability among the given flows. Take Fig. 3 as an example. According to the method in previous section, at least 5 RSUs are needed, such as $S = \{e_2, e_3, e_5, e_7, e_9\}$. However, if C2C communication is allowed, $l_1 = l_3 = l_5 = 0$, $l_2 = l_4 = 1$, and $l_{\max} = 1$, only placing 3 RSUs is enough, where $S' = \{e_8, e_9, e_{10}\}$. Under strategy S', the received tag sets of flows f_1 to f_5 are $\{e_9^{[0]}, e_9^{[1]}\}$, $\{e_8^{[1]}, e_9^{[1]}\}$, $\{e_8^{[0]}, e_8^{[1]}\}$, $\{e_8^{[1]}, e_{10}^{[1]}\}$, and $\{e_{10}^{[0]}, e_{10}^{[1]}\}$. Based on the tags, flows can be securely distinguished from each other. Moreover, our approach establishes a priority level-based requirement: the secure distinguishability among flows with priority level $l_i \leq k$ must be provided by the RSU tags within k-hop. In the example of Fig. 3, the secure distinguishability among flows f_1, f_3, and f_5 can be achieved by purely using the tags directly from the RSUs (i.e. in the form of $e^{[0]}$), and the distinguishability of flows f_1 to f_5 can be achieved by using the tags $\{e^{[0]}\}$ and $\{e^{[1]}\}$. The main idea behind this requirement is that the distinguishability between flows with a higher priority should be provided by more direct, reliable, and credible evidence (i.e. RSU tag.)

B. Problem Formulation for Advanced Model

Realistically, a credential (i.e. $e_i^{[j]}$) from a nearby RSU may not be always available since the tag could be lost or there were not enough cars in certain flows during the multi-hop relays from flow to flow. We denote $p(f_i, e_j^{[k]})$ as the probability that flow f_i receives k-hop propagated tags from the RSU on street e_j. If $e_j \in f_i$, then $p(f_i, e_j^{[k]}) = 1$ for $\forall k \in [0, l_{\max}]$. Let $\mathbb{P}\{\cdot\}$ indicate the probability of any event, and $T(l_i, l_j) \in [0, 1]$ be a predefined threshold for securely distinguishing any two flows with priority levels l_i and l_j, respectively. Note that $T(l_i, l_j)$ is a symmetric non-increasing function: (1) $T(l_i, l_j) = T(l_j, l_i)$; (2) if $l_j < l_k$, then $T(l_i, l_j) \geq T(l_i, l_k)$. Our goal is to deploy a minimum number of RSUs such that the probability for securely distinguishing any pair of flows is no less than a predefined threshold, which is determined by the flows' priority levels. The optimal RSU placement problem with the help of C2C communications is formulated as follows:

$$\min \quad |S| \tag{3}$$

$$\text{s.t.} \quad \mathbb{P}\{S^l(f_i) \not\subseteq S^l(f_j)\} \geq T(l_i, l_j) \text{ for } \forall f_i, f_j \in F \tag{4}$$

where $l = \max(l_i, l_j)$, $S^{[l]}(f)$ denotes a set of received tags that vehicles in flow f can obtain via an exact l-hop C2C credential relay under the RSU deploying strategy S. $S^l(f)$ represents all received tags within l-hop: $S^l(f) = \bigcup_{k=0}^{l} S^{[k]}(f)$. $\mathbb{P}\{S^l(f_i) \not\subseteq S^l(f_j)\} = \mathbb{P}\{S^l(f_i) \backslash S^l(f_j) \neq \emptyset\} = 1 - \prod_{e^{[k]} \in d_{ij}} [1 - p(f_i, e^{[k]})]$, where $d_{ij} = S^l(f_i) \backslash S^l(f_j)$ and $k \leq l$. Eq. 4 requires that the secure distinguishability of flows with priority l must be provided by the RSU-based credentials within l-hop. Clearly, the optimal RSU placement problem in the basic model is a special case of the priority level-based optimal RSU placement problem, where $T(l_i, l_j) = 1$, $l_i = l_j = 0$, $p(f_i, e^{[0]}) = 1$, $p(f_i, e^{[k]}) = 0$ for $\forall f_i, f_j \in F$ and $k > 0$.

C. Problem Analysis

Let f_i^l denote a set of tags, which could be received by the vehicles in flow f_i via at most l-hop C2C relays if all edges were deployed with RSUs. Take f_1 in Fig. 3 as an example. We have $f_1^0 = \{e_1^{[0]}, e_9^{[0]}\}$ since in 0-hop C2C relay vehicles in f_1 can only receive tags directly from the RSUs on e_1 and e_9. Similarly, $f_1^1 = \{e_1^{[0]}, e_1^{[1]}, e_9^{[0]}, e_9^{[1]}, e_2^{[1]}, e_3^{[1]}\}$ since vehicles of f_1 are able to receive not only the tags related to e_1 and e_9 but also the tags from e_2 and e_3 via 1-hop C2C relay.

Using C2C communication essentially increases the coverage of an RSU such that the distinguish sets (i.e. $f_i^l \backslash f_j^l$ and $f_j^l \backslash f_i^l$ for any pair of flows) can be hit easier by a smaller set of selected edges for placing RSUs. For the ease of analysis, we temporarily ignore the probability part and only consider the constraint $S^l(f_i) \not\subseteq S^l(f_j)$ for $\forall f_i, f_j \in F, f_i \neq f_j$, where $l = \max(l_i, l_j)$. We can obtain the following two theorems.

Theorem 4: If f_i and f_j can be securely distinguished, the distinguishability is preserved when using credentials within k-hop, where $k > \max(l_i, l_j)$.

Theorem 5: If strategy S securely distinguishes flows by only using tags directly obtained from RSUs, S also provides secure distinguishability when priority levels are considered.

D. Approximation for Priority Level-based RSU Placement

This subsection presents a greedy approximation solution, Alg. 2 and 3, for the priority level-based RSU placement problem. Intuitively, one may start the approximation by only considering the flows which are in the highest priority level (i.e. $l_i = 0$), and then, gradually including more flows according to the decreasing order of their levels. However, our experimental study shows that its performance is even worse than that of Alg. 1, due to the lack of global coordinations in the beginning phase. In our method, the optimal solution is approximated by gradually solving the problem from the lowest priority level (i.e. $l_i = l_{\max}$) to the highest one.

After initialization (line 1), Alg. 2 creates a candidate set. Here are two options: one is to include all edges of G (i.e. $S_0 = E$), and the other one is to call Alg. 1 and only the edges

Algorithm 2 Priority Level-based RSU Placement (PLRP)

```
1: Initialize $S = \emptyset$
2: Set up candidate edge set $S_0$
3: for each type of priority level $k = l_{\max}, \ldots 1, 0$ do
4:     Create $L'$ by replacing all $l \in L, l < k$ with value $k$
5:     Find RSU placement: $S = RPLK(G, F, L', k, S_0, S)$
6: return $S$ as the RSU placement strategy.
```

Algorithm 3 RSU Placement within Level k (RPLK)

```
1: Initialize flow set within level $k$: $F' = \{f_i | l_i \leq k\}$
2: Initialize distinguish set: $D = \emptyset$
3: for each order of flows $f_i, f_j \in F'$ do
4:     if $\mathbb{P}\{S^k(f_i)\backslash S^k(f_j) \neq \emptyset\} < T(l_i, l_j)$ then
5:         Initialize $d_{ij} = \emptyset$
6:         for every $e^{[k']} \in (f_i^k \backslash f_j^k)\backslash S^k$, $0 \leq k' \leq k$ do
7:             $w = 1 - \mathbb{P}\{S^k(f_i)\backslash S^k(f_j) = \emptyset\}(1 - p(f_i, e^{[k']}))$
8:             $d_{ij} = d_{ij} \cup \{(e^{[k']}, w)\}$
9:         $D = D \cup \{d_{ij}\}$
10: Prune $D$ by removing all supersets
11: for each street, $e \in S_0 \backslash S$ do
12:     Initialize a counter of $C_e = 0$.
13: while $D \neq \emptyset$ do
14:     for each subset, $d_{ij} \in D$ do
15:         for $e \in S_0 \backslash S$ do
16:             if $\exists k' \in [0, k]$ s.t. $(e^{[k]}, w) \in d_{ij}$ then
17:                 Update $C_e = C_e + w$.
18:     Update $S = S \cup \{\arg\max_e C_e\}$.
19:     for $\forall d_{ij} \in D$ related with $\arg\max_e C_e$ do
20:         if $\mathbb{P}\{S^k(f_i)\backslash S^k(f_j) \neq \emptyset\} \geq T(l_i, l_j)$ then
21:             Remove $d_{ij}$ from $D$
22:         else
23:             for $\forall(e^{[k']}, w) \in d_{ij}$ do
24:                 Update $w$ according to current $S$
25:     Reset $C_e = 0$ for each street, $e$.
26: return $S$ as the RSU placement strategy.
```

in the resulting set are included (i.e. $S_0 = SBG(G, F)$) since the solution in the basic model always satisfies the constraint in Eq. 4: $p(f, e^{[0]}) = 1$ if $e \in f$. In Section IV, we study Alg. 2's performance difference of using these two options. From lines 3 to 5, Alg. 2 greedily selects edges from S_0. In each iteration, Alg. 2 temporarily replaces the priority level of flows, which is less than k, with value k (line 4), and then, finds out a set of optimal RSU placing locations for the current setting of priority levels L' (line 5).

Alg. 3 approximates the optimal RSU placement by using C2C communications. It first constructs a flow set $F' = \{f_i\}$ where flows' priority level satisfies $l_i \leq k$ (line 1), and then, initializes the overall distinguish set D in line 2. From lines 3 to 9, Alg. 3 computes the potential tags $e^{[k']}$, which could be used to distinguish flows, and their weights, w. In order to securely distinguish f_i from f_j (line 3), Alg. 3 checks whether the existing strategy S has already satisfied Eq. 4. If it has not, a distinguish set d_{ij} is created. Note that at least one element in the final result S must hit d_{ij}. For any potential tag $e^{[k']}$, which can be uniquely obtained by f_i within k-hop rather than f_j and cannot be provided by its current S (line 6), Alg. 3 computes its hitting probability, w, for providing the distinguishability (line 7) and adds $(e^{[k']}, w)$ to d_{ij} (line 8). Consider that, for sets $d_{ij}, d_{i'j'} \in D$ and $d_{ij} \subseteq d_{i'j'}$, a selected edge e that hits d_{ij} definitely hits $d_{i'j'}$. Before selecting the optimal RSU locations, line 10 removes all supersets (i.e. $d_{i'j'}$) from D.

From line 11 to line 25, Alg. 3 greedily selects the edge with the largest overall hitting weight. For each potential edge that has not be selected (i.e. $e \in S_0 \backslash S$), Alg. 3 creates a counter (line 12) and computes its total weights in D (lines 14-17). In line 18, the edge, e, with the largest weight is selected and added to S. For all d_{ij} containing the potential tags from the edge e (line 19), if it satisfies Eq. 4 (line 20), Alg. 3 removes it from D; otherwise, all the weights in d_{ij} will be updated according to the current S (line 24): $w = 1 - \mathbb{P}\{S^k(f_i)\backslash S^k(f_j) = \emptyset\} \times (1 - p(f_i, e^{[k']}))$ for $\forall(e^{[k']}, w) \in d_{ij}$. Alg. 3 resets $C_e = 0$ for the next iteration (line 25), and it will not stop until all d_{ij} are hit by S.

IV. Evaluation

A. Evaluation Setup and Evaluation Metric

For the ease of description, we call Algorithm 1 *SBG*, and use *RPLK* to represent variants of Algorithms 2 to 3: If only the edges, which are selected by SBG, are included in the candidate set of Algorithm 2, we use *RPLK_S0* to represent it; otherwise, all streets are considered. In practice, priority levels of vehicle flows may, or may not, be predefined depending on the actual applications. If the levels are pre-known, we directly apply Algorithm 2; if the flow levels are undetermined, we let all flows be at level $l_{\max}$, and find the optimal RSU deployment. We use *RPLKM* to represent the situations where priority levels are predefined, and use *RPLK0*, *RPLK1*, and *RPLK2* to indicate the cases that priority levels are undetermined and the value of $l_{\max}$ is 0, 1, and 2, respectively. We not only study the performance of Algorithms 1 to 3 with different inputs, but also compare them with another four baseline algorithms: Coverage-Oriented Greedy (COG), Distinguishability-Oriented Greedy (DOG), Select Unique Coverage (SUC), and Three Stage Placement (TSP). The basic idea of them can be found in paper [5].

We use the number of placed RSUs as our evaluation metric, since the goal of this paper is to securely distinguish given traffic flows by using as few RSUs as possible. Our evaluation consists of both simulations (Figs. 4 to 5) and real data-driven experiments (Figs. 6 to 7). In the simulations, we first create several regions, and then, randomly assign some traffic flows within each region. Among neighboring regions, a set of random traffic flows are further generated. All traffic flows are unique, and there is no flow that is the sub-flow of some other flow. The real data comes from the Dublin vehicle trace and the Seattle bus trace [6]. For the ease of the experiments, we focus on the parts within Dublin's and Seattle's central areas, and all sub-traffic-flows are removed.

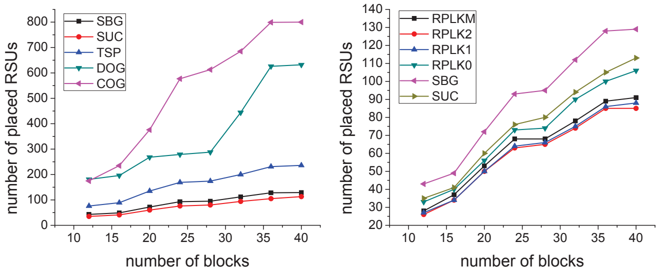


Fig. 4. Comparison among different approximation algorithms

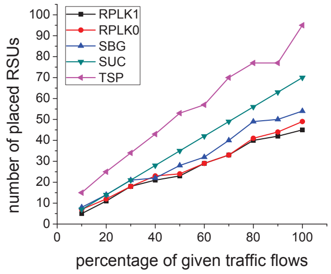


Fig. 6. Dublin bus trace

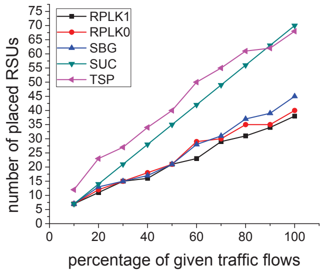


Fig. 7. Seattle bus trace

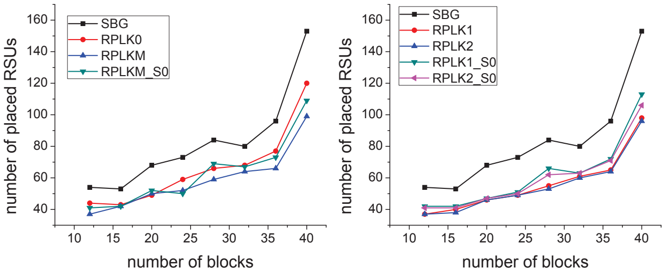


Fig. 5. Impacts of S_0 for Algorithm 2

B. Evaluation Results

Fig. 4 compares the performances of different algorithms. Since their results have huge differences, we plot them on two sub-figures. During the simulation, we gradually increase the number of traffic blocks, which essentially generates more vehicular flows. Since the compared algorithms do not consider the loss of RSU tags, for the fairness of comparison, we assume that there is no package loss in this set of simulations. In Fig. 4, with the growth of the number of traffic flows, all algorithms require an increasing number of RSUs. Methods, DOG and COG, have worse performances than other approaches, and our solution, a series of RPLK algorithms with different inputs, is significantly better than others. The difference between RPLK1 and RPLK2 is very close, which means letting $l_{\max}$ be 1 or 2 is good enough in terms of the computing speed and approximation accuracy. Note that, even without the usage of C2C communications $l_{\max} = 0$, RPLK0 is still better than SBG since there is a special pruning operation in Algorithm 3 line 10.

Algorithm 2 selects locations from a candidate set S_0 for deploying RSUs. S_0 may include all edges of G (i.e. $S_0 = E$) or only the edges selected by Algorithm 1 (i.e. $S_0 = SBG(G, F)$). Fig. 5 studies the impacts of the elements of S_0. Since the performances of the compared methods are very close, we represent the results in two sub-figures. The performances of the RPLK variants are always better than that of the SBG algorithm, no matter what is the initial value of S_0. Since RPLK1_S0, RPLK2_S0, and RPLKM_S0 only consider the partial edges of G for deploying the R-SUs, their performances are worse than RPLK1, RPLK2, and RPLKM. However, their differences are much smaller than the differences between RPLK1, RPLK2, RPLKM, and SBG, respectively. RPLK1 and RPLK2 have a very close performance, which means using 1-hop C2C communication is good enough in terms of the number of required RSUs.

Figs. 6 and 7 are based on real data. During the evaluations, we gradually include more and more vehicular flows. The evaluation result is consistent with that of the simulation. Generally speaking, DOG and COG have the worst performances. With the growth of the number of flows, all approaches need an increasing number of RSUs, but our proposed RPLK algorithm (i.e. RPLK0 and RPLK1) always requires the least number of RSUs. Due to the poor performance of DOG and COG, we do not include their results in Figs. 6 and 7.

V. Conclusion

Vehicular trajectory-based data provides a new perspective for many applications in smart cities. Unlike the conventional data, which is a discrete record, vehicular data is a sequence of spatiotemporal records. Considering that a malicious user may misstate his location claims, it is necessary to generate a location proof for vehicular trajectory-based data. In this paper, such a location proof is created by using the messages from nearby RSUs. We aim to use a minimal number of RSUs to securely distinguish all given traffic flows, which is not a trivial problem. In order to find the optimal locations for deploying RSUs, several algorithms are proposed. Extensive experiments are conducted to evaluate the proposed solutions.

Acknowledgment

This research was supported in part by NSF grants CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, CNS 1439672, CNS 1301774, and ECCS 1231461.

References

[1] Y. Zhang, C. C. Tan, F. Xu, H. Han, and Q. Li, “Vproof: Lightweight privacy-preserving vehicle location proofs,” *IEEE TVT*, 2015.

[2] W. He, X. Liu, and M. Ren, “Location cheating: A security challenge to location-based social network services,” in *ICDCS*. IEEE, 2011.

[3] K. Sohn and D. Kim, “Dynamic origin–destination flow estimation using cellular communication system,” *IEEE TVT*, 2008.

[4] “Full version,” http://people.sju.edu/ wchang/paper/IWQoS17full.pdf.

[5] H. Zheng, W. Chang, and J. Wu, “Coverage and Distinguishability Requirements for Traffic Flow Monitoring Systems,” in *IEEE IWQoS*, 2016.

[6] J. G. Jetcheva, Y.-C. Hu, S. PalChaudhuri, A. K. Saha, and D. B. Johnson, “Design and evaluation of a metropolitan area multitier wireless ad hoc network architecture,” in *WMCSA*, 2003.