29

Environmental-Assisted Vehicular Data in Smart Cities

Wei Chang[1], *Huanyang Zheng*[2], *Jie Wu*[2], *Chiu C. Tan*[2], *and Haibin Ling*[2]

[1] *Department of Computer Science, Saint Joseph's University, Philadelphia, PA, USA*
[2] *Department of Computer and Information Sciences, Temple University, Philadelphia, PA, USA*

CHAPTER MENU

Objectives

- To become familiar with the security and privacy issues of location information in smart cities
- To become familiar with the concept of location proof for trajectories in context of smart cities
- To become familiar with the conventional approaches for generating location proofs and their shortcomings
- To become familiar with the idea of environmental element-based location proof
- To become familiar with the optimal RSU deployment problem, which aims to generate secure and privacy-preserved location proofs by using a minimum number of RSUs

Smart Cities: Foundations, Principles and Applications, First Edition.
Edited by Houbing Song, Ravi Srinivasan, Tamim Sookoor, and Sabina Jeschke.

29.1 Location-Related Security and Privacy Issues in Smart Cities

For the past two decades, the term *smart cities* has gained an increasing attraction from academia, government [1, 2], and industry [3, 4]. Within future smart cities, people are expecting the usage of data, not only from some static pre-deployed roadside sensors but also from intelligent vehicles moving within the cities day after day. A typical intelligent vehicle is equipped with multiple sensing devices, such as on-car cameras and gyroscopes, and also has wireless communication capabilities, such as Wi-Fi/LTE. All these devices record every incidence within the city. Unlike the conventional location-based services or mobile social networks, where data is related with a location spot, the recorded data in intelligent vehicles is a continuous and integrated observation along a vehicle's trajectory (or trajectory segments). In the foreseeable future, these vehicle-based data sequences will support a considerable number of new applications, ranging from criminal scene reconstruction to smart traffic management to environmental monitoring.

The intelligent vehicles will inevitably generate an enormous amount of data. Moreover, the data itself may also bring plenty of security and privacy issues. In order to control the data, a carefully designed data management system is urgently needed. Such a management system must be able to balance the trade-off among privacy, security, and data utility, which is extremely hard. For example, in the application of criminal scene reconstruction, when an incident occurs, how can the data management system efficiently and accurately find all related data, meanwhile providing the privacy of these witnesses? When a driver reports an illegal littering from a vehicle, how does the smart city's system verify that the claimer indeed was at the reported location and time, instead of a frame-up?

In order to preserve data searching privacy, the existing works adopt homomorphism-based data encryption. However, the scheme does not suit intelligent vehicle systems, not only because there is a huge amount of data and searching over cipher text is time consuming, but also because there is no solution for extracting semantic information directly from encrypted images or videos. In addition, from the existing location-based services and mobile social networks, we have already seen the motivations for an adversary to misstate their spatiotemporal claims [5–7]; the encryption-based scheme cannot handle situations in which the data itself is maliciously tampered with by an adversary. Consequently, a key requirement for the intelligent vehicle-involved smart cities involves its abilities (i) to verify the spatiotemporal claims made by a vehicle, (ii) to quickly locate the corresponding queried records from an enormous amount of data, and (iii) to simultaneously provide strong privacy protections to the data owners.

According to certain features of applications, the data owner (i.e., a vehicle and its driver) must be in one of two modes: *proactive* or *reactive*. In the proactive mode, the data owner proactively claims a set of spatiotemporal data, and the data management system should be able to verify these claims, while in the reactive mode, the system searches for the data of the vehicles that are likely to have appeared at a specific location during a specific time period.

In this chapter, we study the use of *environmental factors* to develop "evidence of presence" for the intelligent vehicle system. More specifically, we consider how to use the measured wireless signal from roadside units (RSUs) to *verify* and *index* data. The evidence of presence is a means for a vehicle to demonstrate that it was indeed at a specific location and time. For instance, given a car that claims to have witnessed a particular car collision accident at a specific location at a specific time, we would be able to verify such a claim by comparing the claimer's captured surrounding environmental factors against a known database of environment features. The malicious users, who did not pass by the specific location and time, should be unable to generate the same evidence.

Unlike the existing approaches, where the location proof is constructed by using cryptographic keys, the content of environmental evidences is not linked to the identity of any vehicle, because many applications in smart cities are only interested in the correctness of where and when data is collected. We take a novel approach that relies on measuring the wireless properties inherent in environments, for example, due to the multipath effect, to generate data index (for the reactive mode) or to demonstrate evidence of presence (for the proactive mode). The only task that each vehicle should take is to passively record the surrounding environmental information.

For example, when a vehicle takes a short video at some place and time, the surrounding features (e.g., shadows, colors, brightness, etc.) exhibited in the video frames will be different from those taken at other locations or times, due to the differences caused by environmental factors such as weather condition and random obstructions by physical objects. Similarly, the vehicle moves in a region, and the quality of its received wireless signals from the same transmitter will differ due to factors like interferences and multipath effect. Here, we focus on using wireless signal features to index or verify spatiotemporal data about sequences of observations in vehicle networks. Ideally, when the received wireless signal features at every road stretch are unique, one can easily use the features to index or verify the data of any location at any time. However, in reality, it is too expensive to achieve such a dense coverage on a road stretch. In order to minimize the deploying costs, we further study the optimal placement problem of the roadside signal transmitters and the synchronization problem among different transmitters.

The contributions of this chapter are as follows. We propose a novel approach by exploring the spatiotemporally varied environmental signals to index or verify vehicle networks' data. We also provide an approximation algorithm to

provide a near-optimal placement of the signal transmitters in this system. Finally, we also study the time synchronization issue among different signal transmitters.

29.2 Opportunities of Using Environmental Evidences

Due to the existence of malicious users, every piece of spatiotemporal data should be verifiable by authorities. For instance, when a car accident occurs, the police should not only verify the evidence of presence of a witness (i.e., location claimer) but also check how well the claimer's provided information corroborates with additional evidence, such as the data records of nearby vehicles, surveillance cameras, and environmental factors.

The evidence of presence can be verified via either direct witness (DW) or indirect support (IS). The DW comes from the directly recorded location proofs from nearby attestors, the construction of which is the core of the conventional cryptographic key-based approaches. Considering a group of nearby vehicles, whenever one of them wants to create location-based data, all these vehicles need to exchange some encrypted and spatiotemporal-bounded messages to build the location proof. Although this type of scheme provides high-level security protection, it inevitably discloses the nearby vehicles' location privacy during verification, since each cryptographic key is uniquely linked to a vehicle. In addition, there are also key management issues, such as the revocation and renewal of certain keys.

Considering that not all applications in smart cities need to know the identities of data owners, in this chapter, we construct the evidence of presence for each spatiotemporal *data* rather than vehicles. We use the impacts of some unpredictable environmental factors on the recorded data as IS for the evidence of presence: the adversary, who did not physically appear at the claimed location during the claimed time, is not able to generate the data with the corresponding "environmental marks." The IS-based verification is conducted by checking the consistency of spatiotemporal data's embedded environmental factors against a known database of historical environment features. Admittedly, the IS-based evidence of presence cannot provide a security protection as high as DW does. But it can successfully hold back the attackers who easily make location claims without any physical appearance. Generally, at least six environmental facts can be used as IS:

- *Environment signals*: The control messages of the received environment signals are unpredictable, by which only the vehicles that have physically appeared at that location at that time can possess the data. Moreover, due to the multipath fading and shadowing conditions, the received signals' qualities can also be considered.

- *Road patterns*: The claimer and attestors are driving on the same road segment (straight road, right curve, or left curve) and therefore should have the same turning pattern. Based on the readings of a gyroscope, one can discriminate the cases of driving on a curve from changing lanes [8] and extract the corresponding road patterns in that period.
- *Non-overlapping trajectories*: Since each vehicle takes a space, the claimer and attestors' trajectories should not overlap with other vehicles' trajectories in a spatiotemporal domain.
- *Local co-viewing*: The claimer and attestors are on the same road segments and, therefore, should have similar local views, such as the same front cars or similar nearby scenes, in their camera videos. We also consider the imperfect recording conditions, such as bad weather and unpaved roads, and use them to check the existence of inconsistency.
- *Landmark co-viewing*: Police can also find other vehicles that are at different locations but relatively close to the reported region. From the vehicles' carrying cameras, police may be able to extract the unique random statuses of some landmarks and use them as the indirect supporters of a location claim.
- *EZpass-based PO*: On roadside, there are some randomly deployed EZpass readers (i.e., POs). The reader cannot obtain each vehicle's account number due to the inference, but we can use it to measure the number of cars that have been passed within a short period of time [9]. If the reported number of attestors differing from the reader's measured number is greater than a threshold, the claimer's statement should not be accepted.

However, the randomness of the environmental factors may not be able to provide full distinguishability among a given set of vehicle flows. For instance, if one solely uses weather conditions as an IS-based evidence of presence, the granulite must be at least on a city level. In other words, if the vehicle flows do not consist of the paths through several far away cities, the weather condition-based evidence becomes useless since every vehicle in a city is very likely to experience the same weather. In this chapter, we focus on using wireless signal features to index or verify spatiotemporal data in vehicle networks. Besides the existing cellular towers and Wi-Fi access points, we intentionally deploy several RSUs (i.e., wireless signal transmitters) on certain road stretches and let them generate spatiotemporal-bounded random signals. The signals from both the RSUs and the existing wireless network infrastructures will be used as environmental evidence of presence in our system.

29.3 Challenges of Creating Location Proofs

Location verification, also known as location proof [10], is a well-known problem in the mobile computing communities. The goal of location verification

is to securely prove that a claimer has indeed appeared at a specific location at a specific time. For verifying the spatiotemporal claims, different types of schemes are designed. Using the distance-bounding protocols [11–13] is a common approach, which measures the physical times/distances for messages to transmit between a claimer and verifiers and estimates the claimer's real physical location based on these times. In this type of solution, the verifiers could be other participants, such as mobile phone users and vehicles [14], or some special infrastructures [15]. However, the accuracy of the distance-bounding approaches relies on the deploying density of the verifiers and their trustworthiness.

Cryptographic key-based approach [16–18] is another popular way to generate the location proof, where the claimer and verifiers share a set of spatiotemporal-bounded messages. Although this kind of approach avoids the deployment issues with certain measuring infrastructures, it still has trustworthiness and key management issues.

Recently, people have begun to consider using unique impacts of environmental factors on surrounding objects to create evidence for location verification. Unlike the conventional schemes, the environment-based approaches [19, 20] do not require storing of any cryptographic keys/certificates nor do they require the participants to perform any cryptographic processing, which is very time consuming. Instead, claimers only need to capture some environmental features, such as received signal strength (RSS) or the control messages in 802.11/4G LTE networks, which will be verified later against a known database of features to establish the validity of local claims. Note that all these schemes focus on built evidences for verifying the physical presence on a single location spot.

However, for the vehicular data, it is essentially a sequence of records about the surroundings, from the last data uploading location to the next one. Due to the fact that, for certain applications like criminal scene reconstruction, no one knows which piece of information is useful at the time of recording, we need to create a set of location verifications for the vehicle. Clearly, directly adopting the existing schemes for single spot is too expensive since a vehicle would frequently create plenty of location proofs. The proposed system in this chapter has been inspired by an indoor-tracking paper [21], where authors use a set of collected Wi-Fi data to associate identities with different moving objects in surveillance videos. More specifically, we verify the presence of a vehicular trajectory by providing spatiotemporal-bounded messages only on some crucial road stretches, the combination of which can uniquely distinguish a trajectory from others. From the consideration of computing complexity, our system adopts the RSS-based environmental evidence scheme [20] to generate the messages on crucial road segments, and only the roadside infrastructures possess keys, instead of vehicles. In order to make the vehicular data indexable

and privacy preserving, we embed location and environment information into some time-bounded random numbers and use them as both the index and location proof of the vehicular data. Unlike the time-bounded random numbers generating approach in paper [18], our system's random numbers are bounded to certain preknown locations (i.e., the physical locations of RSUs), and our random numbers are more secure even if the initial random number generating parameters is obtained by attackers. Note that exploring roadside infrastructures is a commonly used approach in smart cities. However, the existing works [22–24] mostly focus on the improvement of data transmissions by using RSUs, while this chapter considers how to use the roadside infrastructures to securely verify/index trajectory data.

29.4 Environmental Evidence-Assisted Vehicular Data Framework

29.4.1 System Model and Attack Model

Our intelligent vehicle-based smart city system consists of three components: vehicles, roadside infrastructures, and a supporting data management system. The proposed system integrates the existing devices on a vehicle and provides a more comprehensive description of a city. We assume that each vehicle is equipped with a video camera, which keeps recording all surrounding events; an EZpass tag, which is associated with a driver's account number; and multiple sensing and communication devices, such as a gyroscope, accelerator, and wireless signal transmitter. Without loss of generality, we use D_i to represent a piece of data. Note that, in our model, a single vehicle can return zero or multiple pieces of data, and how to use the data is determined by the smart city applications, which is out of this chapter's scope.

The centralized data management system is responsible for collecting, searching, and verifying the data pieces D collected from vehicles. Each D_i is implicitly associated with certain temporal and spatial information, (T_i, L_i). In order to protect the location privacy of a data owner and provide the capability of spatiotemporally verifiable evidence of presence, we embed the time and location information into environmental wireless signals (i.e., $\mathcal{E}(\cdot)$) and use them as both data index and verification evidence. Each vehicle only needs to send the location claims to the data management system in the form $(\mathcal{E}(T_i, L_i), D_i)$ rather than directly and explicitly uploading the (T_i, L_i) to the data collector.

For constructing the spatiotemporal-embedded wireless signals, we consider two types of roadside infrastructures: the existing wireless communication infrastructures and some RSUs, which are specially installed by the smart

cities. RSUs are wireless transmitters, and the only task they conduct involves continually broadcasting certain specially designed random signals to the passing vehicle.

There are two types of attackers: privacy prier (PP) and fake claimer (FC). The objective of PP is to establish a connection between a vehicle and its reported data without physically dogging the victims. In this chapter, we focus on the scenarios, where PP tries to find others' location privacy by querying the data management system with some well-designed spatiotemporal query demands.

We also assume that there are a small number of malicious vehicles controlled by FC, who is able to manipulate any value of the controlled vehicles. According to the exact applications, reporting fake spatiotemporal data can be beneficial to the adversary in different ways. For example, in the application of smart traffic management, FC can maliciously create an illusion of having several accidents on a road stretch such that the routing paths may be recalculated or the traffic lights may falsely adjust their switching frequencies or lengths. For the crime scene reconstruction, FC may use some tampered data to frame some victims or exculpate outlaws. In this chapter, we want to prevent the attackers who make fake location claims without any physical presence at the claimed location during the claimed time.

29.4.2 Roadside Unit-Based Environmental Evidence Construction

The construction of the embedded signals is based loosely on [20]. Let us assume that RSU and the central data management system use public/private keys. Here, we only consider the keys of RSU and the data collectors, instead of individual vehicles, the number of which is significantly greater than that of RSU. The physical location of RSU is preknown by the data management system, and a special control message will be randomly generated and sent from the data management system to the corresponding RSU. The control message to RSU_i contains a future time T_0, an initial value u_i, and an increment Δu_i. At run-time T, when a passing vehicle is detected at RSU_i, the RSU randomly selects a transmission power $p > 0$ and uses this signal power to broadcast certain spatiotemporal-embedded messages to the vehicle. Based on these four variables, the corresponding RSU will generate a series of time-dependent random numbers: from future time T_0, each moment T will be represented as $R_i(T) = u_i + \Delta u_i \times \sum_{T_0}^{T} p$. Instead of explicitly using the time value T, our system will take the random number $R_i(T)$ as a time indicator; only the data management system and the corresponding RSU_i can extract the spatiotemporal information from it.

The spatiotemporal message is defined as

$$M_i(T, L) \leftarrow \langle \text{RSU}_i, R_i(T), \text{Enc}_i(p), H_i(T, p) \rangle$$

where $\text{Enc}_i(\cdot)$ represents an encryption by using RSU_i's key, and $H_i(T, p)$ is a hash signature of $(\text{RSU}_i, R_i(T), \text{Enc}_i(p))$. Upon receiving $M_i(T, L)$, a vehicle first

measures the RSS and then constructs evidence of presence as the following:

$$\mathcal{E}(T, L) \leftarrow \langle M_i(T, L), \text{RSS} \rangle$$

In the proactive mode, a user can make a location claim by $\langle T, L, \mathcal{E}(T, L), D \rangle$, and then, the data management system will verify whether the claimed location and time is consistent with $\mathcal{E}(T, L)$. In the reactive mode, users stochastically upload a sequence of data records, $\langle \mathcal{E}(T, L), \{D\} \rangle$, to the data management system whenever they have Wi-Fi access.

29.4.3 Environmental Evidence-Assisted Application Models

29.4.3.1 Location Claim Verification

The environmental factor-based evidence of presence provides a strong protection against FC: for an individual attacker, unless the keys of both data management system and RSU are compromised at the same time, any fake location claim can always be identified.

The verification has two phases. The first phase is a simple filtering, which simply checks the information consistency within message $\mathcal{E}(T, L)$. The system first verifies whether the claimed local L is under the signal range of the reported RSU_i. Next, it extracts the cipher text $\text{Enc}_i(p)$ from the claimed message, decrypts it by using RSU_i's key, and compares the result with the reported RSS value. If they match, it is likely that the claimer had been physically present at the claimed time and location.

However, considering that the number of possible power levels is very limited, there is an extreme situation that an adversary may correctly guess the value of p at some moments. To solve the situation, regarding a local claim that has passed the first phase of verification, the data management system conducts a second round of verification, which requires direct communication with RSU. Upon finishing the first phase of verification, the data management system further checks the data consistency between the reported random number $R_i(T)$ and RSU_i's historic records. The system requires RSU_i to send a list of its selected transmitting powers from T_0 to the claimed time T, reconstructs the spatiotemporal-bounded random number as $R_i'(T)$, and compares it with the reported value $R_i(T)$. If no inconsistency is detected during this phase, the location claim is trustworthy, unless the adversary compromises both RSU and the data management system.

29.4.3.2 Privacy-Preserved Data Collecting

In the conventional location proof for mobile users, a user intends to prove his physical appearance at a location spot in a moment. However, for many applications in smart cities, including on-car camera-based city surveillance, crime scene reconstruction, searching for abducted children, and smart traffic controls, any spot on a vehicle's trajectory may contain critical information,

which is unknown at the time of recording. Therefore, along the moving path of a vehicle, a series of on-road records will be generated, and the whole data segment will be stored as a data unit on a server. In the reactive mode, the crux becomes how to use environmental evidence to index and retrieve the data about a period of walking in a privacy-preserved way, which directly affects the efficiency of the system.

For the design of a privacy-preserved data collecting/searching system, the trade-off between users' privacy and data utility is an important issue: on the one hand, when an accident occurs, the surveillance system should be able to quickly identify the accident's witness and the corresponding location proofs, which are the evidences showing that the witness was indeed at the region near the accident; on the other hand, the surveillance system must also consider the privacy of individual users, whose historical visiting sequences must be hidden. Based on this trade-off, our surveillance system separately stores the index of a vehicle's historical data from the whole data.

In this chapter, we build a multiagency privacy-preserved system, where different agencies are unable to see the content of any vehicle without the cooperation of others. Basically, there are three components: (i) a data management server, which is responsible for collecting, searching, or verifying spatiotemporal data; (ii) a special storage system for supporting smart city applications, which consists of several clouds; and (iii) a bookkeeping server, which stores the mapping between each data record and its access address in the clouds.

For the users in reactive mode, they stochastically upload their recorded surrounding data, from the last uploading time to the current time onto their own selected clouds in the form of $\langle \mathcal{E}(T, L), \{D\} \rangle$. Note that the users do not need to provide any information about themselves during uploading. For each record $\langle \mathcal{E}(T, L), \{D\} \rangle$, it must be associated with one environmental evidence $\mathcal{E}(T, L)$, and data $\{D\}$ are partitioned into different segments according to the closest $\mathcal{E}(T, L)$. Upon receiving the record, the cloud returns the corresponding access address A to the user. Clearly, only the vehicle's owner knows the full access addresses of his data. Next, the user sends A to the bookkeeping server, and the server will generate a unique random number R and send it back. Data pair (R, A) is locally saved in the bookkeeping server. For the last step of data collecting, the user sends the indexing $\langle \mathcal{E}(T, L), R \rangle$ to the data management server. Figure 29.1 shows the structure of our environmental evidence-based data collecting system. Since the data collecting process does not involve any user's identity, our system is privacy preserved.

For the reactive mode, a data management server essentially is an index server, and all indexes are sorted according to the values of RSU_i and $R_i(T)$ in $\mathcal{E}(T, L)$. Note that for each initial value pair (T_0, u_i), the values of the following time-dependent random numbers are strictly increasing, which partially reflects the temporal visiting orders of different vehicles at the same RSU.

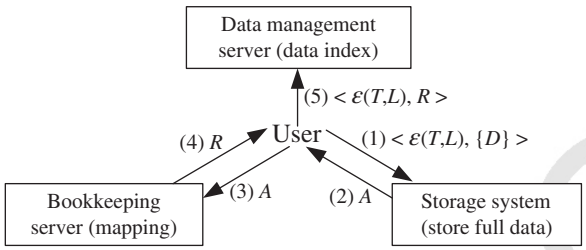

Figure 29.1 Environmental evidence-based data collecting process.

29.4.3.3 Environmental Index-Based Data Retrieval

The basic environmental index-based data retrieval process is as follows. When an incident occurs at $\{T, L\}$, the law enforcement will query the data management server by using $\mathcal{E}\{T, L\}$, and the server will return a set of record numbers $\{R\}$, whose environmental indexes match the query content. Based on the return results, the law enforcement will contact the bookkeeping server and find the physical storing addresses $\{A\}$. Finally, the data segments can be obtained from the storage system by using $\{A\}$. However, in reality, most incidences do not happen around RSUs. How to find out a set of potential witnesses by using the environmental indexes must be considered, which essentially relates with RSU-based localization.

The main idea of the RSU-based localization is that, at a given past time T, the location of any vehicle can be estimated based on the traveling distances toward one or several RSUs. Let $T_i(\cdot)$ and $L_i(\cdot)$ be the record time and location of RSU_i's environmental evidence, respectively, and let L and T be the vehicle's current location and time. Assume that the vehicle had received a set of evidences from $\{\text{RSU}_i\}$, $1 \le i \le l$, and $T_{i-1} < T_i$. Based on the environmental evidences, the location of the vehicle at T can be pinpointed at the locations satisfying the following set of equations:

$$\| L - L_i \| = \int_{T_i}^{T} s(\tau)\, d\tau, \forall i \in [1, l]$$

where $\| L - L' \|$ is the length of the road from L to L' and $s(\cdot)$ represents the vehicle's historical speed at any moment. Similarly, for every RSU_i near the incident's location, we can compute a time window for the users, who potentially may be witnesses, as the following: $T_i \in [T - \frac{\|L-L_i\|}{S_{\min}}, T - \frac{\|L-L_i\|}{S_{\max}}]$, where T and L represent the time and location of the incident. Take Figure 29.2 as an example. Suppose that a vehicle received two consecutive environmental evidences, respectively, from RSU_c and RSU_e. Let T' be a moment between the receiving times of the tags. Based on the vehicle's speed information, we can compute the traveling distance from RSU_e to the vehicle's location at T'. In Figure 29.2, there are three possible destinations at which the vehicle may arrive from RSU_e based on the distance, and there are two other locations that

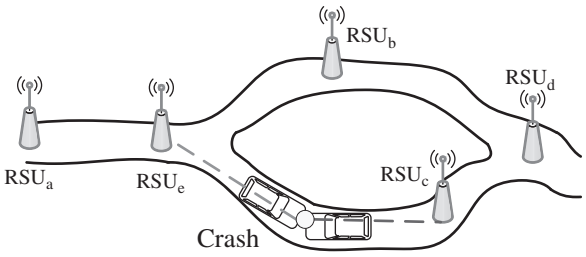

Figure 29.2 The generation of target environmental index.

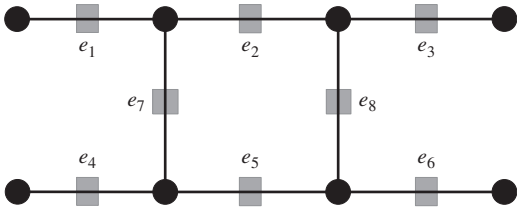

Figure 29.3 Distinguish six vehicle flows in Table 29.1 by RSUs. The black vertices represent intersections, edges indicate road stretches, and the gray boxes give the potential places where an RSU can be deployed.

the vehicle could arrive at RSU$_c$. The intersection of these two sets gives the estimated location at time T'.

29.4.4 Optimal Placement of Roadside Units

In the previous section, our investigations assumed an ideal environment in which sufficient RSUs cover every road stretch in a city. However, in practice, such a situation is unlikely to occur. In order to minimize the deployment costs of the RSUs, a special optimally placing algorithm is needed.

However, this problem is not trivial since not every road segment needs an RSU, and although it is hard for an attacker to forge environmental evidence, the attacker can still easily hide some received evidence in order to pretend that he was somewhere else. Take Figure 29.3 as an example. Assume that there is a map consisting of eight road stretches and there are six vehicle trajectories on the map, which are given in Table 29.1. Our objective is to

Table 29.1 RSU-based tags of given flows in Figure 29.3.

f_1	$e_1 \to e_7 \to e_5 \to e_6$	Ø	e_7	e_7, e_6
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_4, e_3	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_8, e_6
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

install a minimal number of RSUs on some road stretches such that every trajectory can be uniquely identified according to the received environmental evidences. More specifically, considering that some RSUs may use the same transmitting powers or time-dependent random numbers, here we focus on the distinguishability exclusively based on RSU identity numbers within the environmental evidences. For the ease of description, we name the received environmental evidences' RSU identities as *tags*.

Table 29.1 gives three different methods of RSU displacement in Figure 29.3. If only honest users are considered, the optimal RSU placement set is $\{e_2, e_3, e_4\}$, and the received tag sequences of each flow are given in "tags 1" column of Table 29.1. Clearly, all of them have different tag sequences, and therefore, they are fully distinguishable. However, this displacement 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. As a result, when an attacker exists, all flows must be covered by some tags. Column "tags 2" shows an optimal placement by deploying RSUs on stretches e_4, e_7, and e_8, and this placement provides full distinguishability and coverage on the given flows. However, in terms of security, the requirements of full coverage and full distinguishability are not enough. For the attackers who travel along the flow f_6, they are able to be disguised as either f_1 or f_2 by intentionally dropping tags from e_4 or e_7, since the tag sequence of f_6 is a super-sequence of that of f_1 and f_2. The secure and optimal RSU placement in Figure 29.3 is to deploy RSUs on $\{e_4, e_6, e_7, e_8\}$, and the corresponding tag sequence of each flow can be found in the "tags 3" column of Table 29.1's.

Generally, the optimal placement of RSUs must guarantee three conditions. First, a minimal number of RSUs are deployed on certain road stretches. Second, the vehicles traveled along different routes must be distinguishable according to their received environmental evidences' RSU identities. Last, considering that an adversary may intentionally drop certain RSUs' messages in order to create fake location claims, a flow's tag sequence cannot be the subsequence of any other flow's received tag sequence.

29.4.4.1 Problem Formulation

Let graph $G = (V, E)$ denote a map, where node set V is a set of road intersections, and edge set $E = \{e\}$ represents all road segments on G with $E \subseteq V^2$. G contains m predefined vehicle flows $F = \{f_1, f_2, \ldots, f_m\}$. Each flow is represented as a *walk*, which is a sequence of edges, $f_i \in (e_1, e_2, \ldots)$. For the ease of description, we redefine the symbol "$\subseteq$" to represent a subsequence relationship, and we have $f_i \subseteq f_i$, $\emptyset \subseteq f_i$. Note that all flows in F satisfy: $f_i \nsubseteq f_j$ for $\forall i, j, i \neq j$, and in a walk, both nodes and edges can be repeated, as illustrated in Figure 29.4. For instance, in Figure 29.4, if all edges are deployed with RSUs, the tag sequence for f_2 is $(e_1, e_5, e_8, e_6, e_2, e_5, e_8, e_6, e_3, e_4)$, which is a walk.

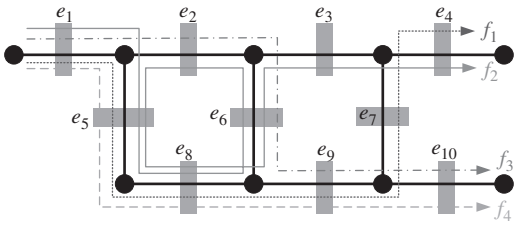

Figure 29.4 Distinguish four vehicle flows by roadside stations.

In order to securely distinguish vehicles of different vehicle flows, several RSUs are deployed on E: whenever a vehicle passes an RSU, the vehicle will receive an environmental evidence, which contains the unique ID of the RSU [25]. Let x_e denote whether road segment e contains an RSU (i.e., $x_e = 1$) or not (i.e., $x_e = 0$) and f' be the road tag sequence of f. f' is a subsequence of f, where only the elements e of f with $x_e = 1$ are kept. We say flows f_i and f_j are *securely distinguishable* if their tag sequences are not in subsequence with each other: $f_i' \nsubseteq f_j'$ and $f_j' \nsubseteq f_i'$.

Our objective is to securely distinguish all flows in F by deploying a minimum number of RSUs on E. The optimal RSU placement problem can be formulated as follows:

$$\begin{aligned} \min \quad & \sum_{e \in E} x_e \\ \text{s.t.} \quad & f_i' \nsubseteq f_j', \quad \forall i, j, i \neq j \\ & x_e \in \{0, 1\} \end{aligned}$$

In the environmental evidences, RSU tags (i.e., RSU_i) show the spatial relationship among different flows [26], and the time-bounded random numbers (i.e., $R_i(T)$) issued from different RSUs offer temporal relationships to indicate the direction of each flow. Note that, in practice, a flow may contain multiple vehicles, and each vehicle will save a series of on-road data records, from the beginning to the end of the flow, as one data unit.

29.4.4.2 Properties

Theorem 29.1 The optimal RSU placement is NP-hard.

Proof: For each pair of flows f_i and f_j, we define a distinguish set as $d_{ij} = \{e_k | e_k \in f_i, e_k \notin f_j\}$, which gives a set of possible locations on which deploying RSUs can distinguish flow f_i from f_j. Due to the requirement of non-subsequence relation, $d_{ij} \neq d_{ji}$. The whole distinguish set for all flows is $D = \{d_{ij}, \forall i, j, i \neq j\}$. We say d_{ij} is *covered* by an RSU placement set $d' = \{e | x_e = 1\}$ if $\exists e \in d'$ such that $e \in d_{ij}$.

The optimal RSU placement problem is to find an optimal set $d^* = \{e|x_e = 1\}$ such that every element $d_{ij} \in D$ is covered by d^*. Clearly, it is a variation of the classic maximum independent set problem [27], which is NP-hard.

Note that the optimal placement of RSUs only under the constraints of full coverage and full distinguishability is also an NP-hard problem [28]. But from the consideration of securities, we must consider the constraint about non-subsequence. As any two flows in F are distinct and any flow is not the subsequence of others, an optimal RSU placement always exists. In the worst case, one can simply install RSUs on every road stretch, and then all flows are securely distinguishable. In addition, we have the following bounds on the minimum and maximum numbers of RSUs.

Theorem 29.2 The minimum number of roadside stations, which can provide distinguishability to F, must be no less than $\lceil \log_2 m \rceil$, where m is the cardinality of F.

Proof: We prove it by contradiction. Suppose there is an optimal placement using $\lceil \log_2 m \rceil - 1$ stations. We give these stations an order and use a binary number with length $\lceil \log_2 m \rceil - 1$ to represent whether a flow received the corresponding tags. There are totally $2^{\lceil \log_2 m \rceil - 1}$ possible values of this number. Let $k = \lceil \log_2 m \rceil$, then we have $2^{k-1} < m \leq 2^k$. Because $2^{\lceil \log_2 m \rceil - 1} = 2^{k-1} < m$, there must exist at least one pair of flows, f_i and f_j, having received a same set of tags. In other words, f_i and f_j are indistinguishable. Contradiction occurs, and therefore, the minimum number of roadside stations should be greater than or equal to $\lceil \log_2 m \rceil$.

The result shows the limit of binary coding to distinguish m flows.

Theorem 29.3 The minimum number of roadside stations, which can provide distinguishability to F, must be no more than $\min\left(\frac{m(m-1)}{2}, |E_F|\right)$, where m is the number of flows and E_F is the edge set of F.

Proof: In the worst case, for every pair of flow, we need to build a new roadside station to distinguish them. Therefore, there are at most $\frac{m(m-1)}{2}$ stations. In addition, for the given flow set $F = \{f_1, f_2, \ldots, f_m\}$, since $f_i \neq f_j, \forall f_i, f_j \in F$, the set of the optimal solution used edges must be a subset of $E_F = \{e|e \in f_i, \forall f_i \in F\}$.

The results show two worst cases: (i) one station is needed to separate every pair of flows for a total number of m flows, and (ii) each edge has one station in place.

Algorithm 1 Distinguishability-Oriented Greedy (DOG) Approximation

1: Construct distinguish set $\{d_{ij}\}$
2: **while** $D \neq \emptyset$ **do**
3: Select one edge $d^* \leftarrow d^* \cup \{e_i\}$ covered most number of sets in $\{d_{ij}\}$
4: Update $\{d_{ij}\}$ by removing the sets which have been covered

29.4.4.3 Approximation for the Optimal RSU Placement

Algorithm 1 is a Distinguishability-Oriented Greedy Algorithm, which always selects the edge covering the most number of elements in the remaining set. However, unlike the maximum independent set problem, in the optimal RSU placement problem, the constructed tag sequence of each flow cannot be the subsequence of any other flow's tag sequence. To approximate the optimal result, we first construct the overall distinguish set D for all flows, which is given by Algorithm 2, lines 3–5. Consider that for the flows satisfying $d_{ij} \subseteq d_{i'j'}$, when an optimal set d^* covers d_{ij}, it must also cover $d_{i'j'}$. In other words, the RSU placements satisfying d_{ij} must also provide both distinguishability and coverage for $d_{i'j'}$. Therefore, in Algorithm 2, lines 7–10, we eliminate the subsequence relations from D. For the remaining elements of D, assuming d_i, if it contains one and only one edge, then this edge e must be associated with an RSU; otherwise, the corresponding flows related with the d_i will not be securely distinguished or fully covered. We call these types of edges *requisite edges*, and lines 12–17 create the optimal RSU placement set d^* by including all requisite edges. The overall distinguish set D is updated by eliminating all elements that are covered by the constructing set d^*. Finally, from lines 20 to 26, within the resulting set D, we construct d^* by greedily selecting the edges covering the largest number of remaining elements in D. The process stops when all elements of D are covered by d^*, and d^* gives the approximated optimal locations for placing RSUs.

Let's consider an example in Table 29.1 and Figure 29.3. For the six flows, their pairwise distinguish sets are shown in Table 29.2. We eliminate any d_{ij} from D if $\exists d_{i'j'} \in D$ s.t. $d_{i'j'} \subseteq d_{ij}$. The resulting $D = \{\{1,7\},\{4\},\{7,5\},\{2,8\},\{6\},\{8,3\},\{5,8\},\{7,2\}\}$. Since some d_{ij} values only contain e_4 or e_6, they are the requisite edges of the given flows. Therefore, at the third part of Algorithm 2, we create $d^* = \{4,6\}$ and update D to $\{\{1,7\},\{7,5\},\{2,8\},\{8,3\},\{5,8\},\{7,2\}\}$. Find the remaining edges in D, which are $E_D = \{1,2,3,5,7,8\}$, and compute the edges' appearing times in D: $|Q(e_1)| = 1$, $|Q(e_2)| = 2$, $|Q(e_3)| = 1$, $|Q(e_5)| = 2$, $|Q(e_7)| = 3$, and $|Q(e_8)| = 3$. e_7 and e_8 appear the most times; we randomly select e_7, and d^* becomes $\{4,6,7\}$. Updating D, E_D, and $Q(e)$, we have $D = \{\{2,8\},\{8,3\},\{5,8\}\}$, $E_D = \{2,3,5,8\}$, $|Q(e_2)| = 1$, $|Q(e_3)| = 1$, $|Q(e_5)| = 1$, and $|Q(e_8)| = 3$. Since e_8 has the highest appearing frequency, we put another RSU on e_8, $d^* = \{4,6,7,8\}$. After another round of updating, D becomes an empty set and Algorithm 2 terminates. The final optimal edges for deploying RSUs are e_4, e_6, e_7, and e_8.

Algorithm 2 RSU Optimal Placement Approximation

```
1: D ← ∅, d* ← ∅
2: /* Construct distinguish set: lines 2-5 */
3: for ∀f_i,f_j ∈ F do
4:    d_ij ← {e|e ∈ f_i, e ∉ f_j}, d_ji ← {e|e ∈ f_j, e ∉ f_i}
5:    D ← D ∪ {d_ij, d_ji}
6: /* Eliminate subsequence relation: lines 6-10 */
7: Sort D = {d_1, d_2, ..., d_k} s.t. |d_i| ≥ |d_j| if i < j
8: for i ← 1 ... k do
9:    if ∃d_j ⊆ d_i, j > i, d_j ∈ D then
10:      D ← D\{d_i}
11: /* Include requisite edges in d*: lines 11-17 */
12: for ∀d_i ∈ D do
13:    if |d_i| == 1 then
14:       d* ← d* ∪ d_i, find e s.t. e ∈ d_i
15:       for ∀d_j ∈ D do
16:          if e ∈ d_j then
17:             D ← D\{d_j}
18: /* Find other elements of d* by a greedy scheme: lines 18-26 */
19: while D ≠ ∅ do
20:    Create edge set E_D ← {e|∃d ∈ D, e ∈ d}
21:    For ∀e_k ∈ E_D, compute Q(e_k) ← {d|∃d ∈ D, e_k ∈ d}
22:    Find e_i ∈ E_D s.t. |Q(e_i)| ≥ |Q(e_j)| for ∀e_j ∈ E_D, i ≠ j
23:    d* ← d* ∪ {e_i}
24:    for ∀d_j ∈ D do
25:       if e_i ∈ d_j then
26:          D ← D\{d_j}
```

Theorem 29.4 By deploying RSUs on the edges found by Algorithm 2, the tag sequences of any two flows $f_i, f_j \in F$ surely satisfy $f_i' \not\subseteq f_j'$.

Proof: We prove the theorem by contradiction. Assume that there is at least one pair of flows $f_i, f_j \in F$, whose tag sequences satisfy $f_i' \subseteq f_j'$. There are totally three conditions that may cause a subsequence relation: (i) $f_i' = \emptyset$, (ii) $f_i' = f_j'$, and (iii) $f_i', f_j' \neq \emptyset, f_i' \neq f_j', f_i' \subseteq f_j'$. For the given flow set F, any two flows are unique and non-sequence $f_i \not\subseteq f_j$, and, therefore, there is at least one pair of edges e_i and e_j satisfying $e_i \in f_i$, $e_i \notin f_j$, $e_j \in f_j$, and $e_j \notin f_i$. According to the definition of distinguish sets, we have $d_{ij} \cap d_{ji} = \emptyset$, $e_i \in d_{ij} \neq \emptyset$, and $e_j \in d_{ji} \neq \emptyset$. Since Algorithm 2 requires $d^* \cap d \neq \emptyset$ for $\forall d \in D$, the resulting set d^* must contain edges e_i^* and e_j^* such that $e_i^* \in d_{ij}$ and $e_j^* \in d_{ji}$, which also means that neither f_i' nor f_j' can be empty; it is impossible for condition (i) to occur. Since $d_{ij} \cap d_{ji} = \emptyset, f_i'$ must possess at least one tag e_i^*, which f_j' does not contain. So,

Table 29.2 The construction of d_{ij} for flows in Figure 29.3.

d_{ij}	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$
$i=1$		$\{1,7\}$	$\{1,7,6\}$	$\{7,5\}$	$\{6\}$	$\{1,5,6\}$
$i=2$	$\{4\}$		$\{6\}$	$\{4,5\}$	$\{4,6\}$	$\{5,6\}$
$i=3$	$\{4,8,3\}$	$\{8,3\}$		$\{4,5,3\}$	$\{4\}$	$\{5,8\}$
$i=4$	$\{2,8\}$	$\{1,2,8\}$	$\{1,2,6\}$		$\{2,6\}$	$\{1,8,6\}$
$i=5$	$\{8,3\}$	$\{1,7,8,3\}$	$\{1,7\}$	$\{7,5,3\}$		$\{1,5,8\}$
$i=6$	$\{4,2,3\}$	$\{7,2,3\}$	$\{7,2\}$	$\{4,7,3\}$	$\{4,2\}$	

both conditions (ii) and (iii) cannot happen. As a result, by using Algorithm 2, any two flows must satisfy $f_i' \not\subseteq f_j'$.

29.4.4.4 Extension: Optimal RSU Placement with Package Loss

In the real world, moving objects such as trucks can block the communication line of sight between an RSU and cars [13]. Therefore, wireless signal-based environmental evidences may fail to be delivered to the passing vehicles. Missing an RSU's signal may cause location verification or the data retrieval of a vehicle flow to fail. For example, in Figure 29.2, losing the tags from RSU_b or RSU_c will cause the system to be unable to determine whether the corresponding vehicles were traveling along the upper or the lower path, especially when the paths have a similar length.

The package loss rates on different road stretches may not be the same due to their traffic densities and topographies. We denote r_i as the package loss rate for each RSU that is placed on the road stretch e_i, and b_i is used to represent the billing (i.e., cost) for deploying an RSU on e_i. Multiple RUSs can be placed on the same road stretch to mitigate package losses. For vehicles, receiving several RSUs' environmental evidences on a road stretch is functionally equivalent to obtaining a single tag in the ideal model, where package loss rate is zero. Moreover, we assume that the tag losses for different RSUs are independent of each other. Let $k_i \in \{0, 1, 2, \dots\}$ denote the number of RSUs placed on the road stretch e_i. Then, the probability of the tag delivery on e_i can be calculated as $1 - r_i^{k_i}$, and the corresponding deploying cost is $b_i \times k_i$. Since vehicles may receive multiple environmental evidences on one road stretch, we redefine the road tag sequence as the following: f' is a subsequence of vehicle flow f, where only the elements e_i of f with $k_i > 0$ are kept.

The problem of optimal station placement with tag loss is defined as follows: using a minimal RSU deployment costs such that every different vehicle flow is theoretically distinguishable and that the average recognizing probability on-road stretches, on which RSUs are deployed, is no less than a predefined threshold. The problem of optimal station placement with tag loss can be reformulated as the following:

$$\begin{aligned} \min \quad & \sum_{e_i \in E} (b_i \times k_i) \\ \text{s.t.} \quad & f_i' \not\subseteq f_j', && \forall i, j, i \neq j \\ & 1 - r_i^{k_i} > \tau, && \forall k_i \neq 0 \\ & k_i \in \{0, 1, 2, \dots\}, && \forall e_i \in E \end{aligned}$$

The approximation algorithm for the optimal RSU placement problem with package loss is given in Algorithm 3. Since the requisite edges must be deployed with RSUs in order to provide a full distinguishability, the beginning parts of Algorithms 2 and 3 are the same. However, for the construction of the remaining part, Algorithm 2 always selects the edge covering the most number of distinguish sets in the remaining D, while Algorithm 3 picks the one with the least cost per set coverage. Algorithm 3 line 4 computes the total costs $B(e_k)$ for achieving the required RSU-based recognizing probability on edge e_k. In line 5, it finds out the distinguish sets $Q(e_k)$ that would be covered after the edge e_k has been selected. The final set d^* is constructed by using the edge with the lowest $B(e_i)/|Q(e_i)|$ value, where $|\cdot|$ is the cardinality of a set.

29.4.4.5 Performance Analysis

In this section, we test the performance of the RSU placement algorithms, Algorithms 2 and 3. We use Figure 29.3 as the regional map, which consists of 8 road stretches, and use the 6 flows in Table 29.1 as the given flows within the region.

We first test Algorithm 2. First, we consider the impotentness for the deploying locations of RSUs. In Figure 29.5, we gradually increase the number of RSUs, which are deployed within the given region and compare the difference of RSU tag-based distinguishability by using the deploying strategies of Algorithm 2 and a random approach, which randomly selects several road stretches to install RSUs. In order to measure the distinguishability among flows, we propose a concept, called securely distinguishable rate (SDR). Recall that, for any pair of vehicular flows f_i and f_j, if their RSU tag sequences satisfy $f_i' \not\subseteq f_j'$, then we say f_i and f_j are securely distinguishable. Similarly, SDR computes the percentage of securely distinguishable flow pairs out of all possible pairs, and

Algorithm 3 RSU Placement with Package Loss

1: Algorithm 2 lines 1-17.
2: **while** $D \neq \infty$ **do**
3: Create edge set $E_D \leftarrow \{e | \exists d \in D, e \in d\}$
4: For $\forall e_k \in E_D$, compute $B(e_k) \leftarrow b_i \times \lceil \frac{\log(1-\tau)}{\log r_i} \rceil$
5: For $\forall e_k \in E_D$, compute $Q(e_k) \leftarrow \{d | \exists d \in D, e_k \in d\}$
6: Find $e_i \in E_D$ s.t. $\frac{B(e_i)}{|Q(e_i)|} \leq \frac{B(e_j)}{|Q(e_j)|}$ for $\forall e_j \in E_D, i \neq j$
7: $d^* \leftarrow d^* \cup \{e_i\}$
8: **for** $\forall d_j \in D$ **do**
9: **if** $e_i \in d_j$ **then**
10: $D \leftarrow D \backslash \{d_j\}$

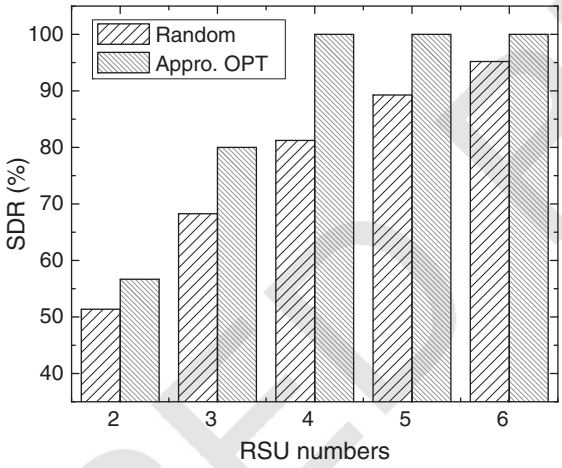

Figure 29.5 Securely distinguishable rate (SDR).

simulation results are shown in Figure 29.5. From the figure, we can see that the SDR values of both random scheme and Algorithm 2 are initially very close to each other when only few RSUs are used; however, with the growth of the RSU numbers, the SDR of Algorithm 2's deployment significantly and quickly goes up to 100%. Since different road stretches possess diverse impotentness for flows' distinguishability, the deploying locations of RSUs must be carefully selected.

In Figure 29.6, we randomly assign a tag loss rate to each road stretch and check the impact of RSU construction costs on the deployment locations. In this part of simulation, we first let all stretches have the same deploying costs, and then, we intentionally add some random extra costs on the critical edges, which are selected by Algorithm 2. The average extra costs are 25% and

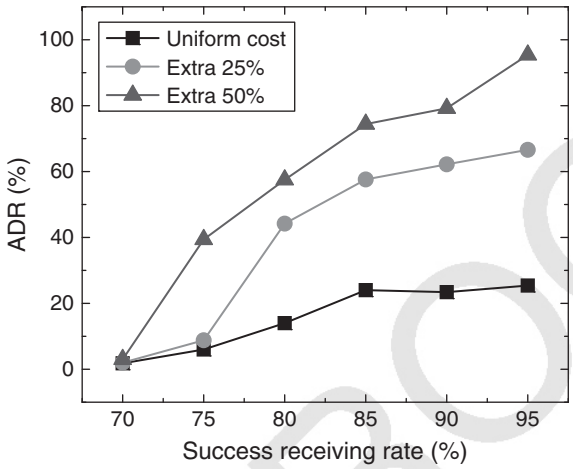

Figure 29.6 Alternate deploying rate (ADR).

50%, respectively. For observing the change of the deployment locations, we further propose another concept, called alternate deploying rate (ADR), which counts the percentage of Algorithm 3's results using an alternate deployment other than the previous Algorithm 2's result. The greater the ADR is, the more impacts of construction costs on the RSU deployment. In Figure 29.6, we gradually increase the minimal tag acceptance rate τ and the ADR values under different construction costs. Figure 29.6 clearly shows that, with an increasing τ, more and more cases drop the previous deploying result (i.e., Algorithm 2) and turn to use some cheaper stretches for achieving a fully secure distinguishability.

29.4.5 Time Synchronization among Roadside Units

In reality, some RSUs, especially the ones deployed in less-traveled regions, may not be able to access the Internet. For using the environmental evidence-based verifiable data indexing in smart city, some special cars are used to periodically collect the historically used RSS time sequence $\{p\}$ from these RSUs and reassign the random number-related parameters $(T_0, u_i, \Delta u_i)$ to them. Although the verification process of the evidence of presence in these regions takes more time, the whole environmental evidence-based system works normally.

The functionality of our system is based on a crucial assumption that all RSUs are time synchronized. For most regions, this requirement can be easily achieved, as long as there are Internet connections or GPS signals. In practice, RSUs are usually cheap devices without high-accuracy atomic clocks [29]. So, there may be time drifting issues for the RSUs without any network connection, which inevitably results in time inconsistency among vehicles and RSUs. Recall that, for avoiding an environmental evidence being modified by attackers, the

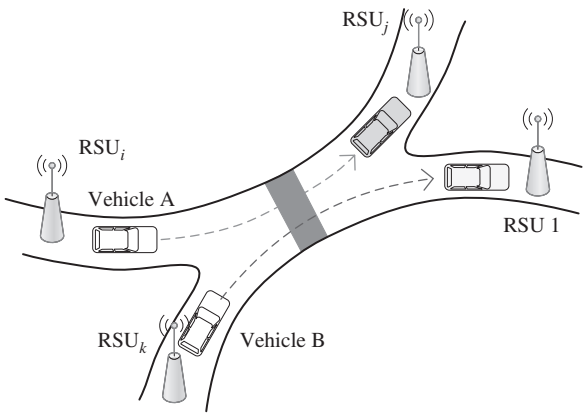

Figure 29.7 Clock synchronization problem among RSU stations. All four roadside stations are not synchronized. The system can never tell whether vehicle A or B passed the shadowed region first.

RSUs' messages are associated with a stations' own time-bounded random number. When the local clocks of RSUs are unsynchronized, not only the data management server is unable to verify the authenticity of some location claims in the proactive mode but also the inconsistency can cause data ambiguities or disorder in the reactive mode.

Take Figure 29.7 as an example. First, the numerical relationship of two RSUs' time stamps may cause disorder in the interpretation of the real visiting sequence. Suppose that the vehicle A consecutively received two environmental evidences $\mathcal{E}\{\mathrm{PC}_a, L_a\}$ and $\mathcal{E}\{\mathrm{PC}_c, L_c\}$, respectively, from RSU_a and RSU_c at times T_a and T_c, where PC_i represents the physical clock of RSU_i. Since the car moved from RSU_a to RSU_c, the real generation time of the evidences must satisfy $T_a < T_c$. However, due to unsynchronized time of RSU_a and RSU_c, the time information embedded in the evidences may become $\mathrm{PC}_a > \mathrm{PC}_c$, which could be interpreted as A driving from RSU_c to RSU_a. Second, the inconsistent local clocks make data incomparable. Also in Figure 29.7, assume that RSU_a and RSU_c are synchronized and so are RSU_b and RSU_d, but RSU_a and RSU_b are not. Vehicle A moved from RSU_a to RSU_c, and vehicle B drove from RSU_b to RSU_d. However, since the time is unsynchronized, we cannot determine whether A or B passed the shadowed region first. In real life, the passing order of a region is critical in criminal investigations. Therefore, the clocks of RSUs must be synchronized.

Here, we use special vehicles to periodically synchronize the local clocks between different RSUs. For each RSU, if no vehicle passes RSU_i at physical time t, then PC_i is differentiable at t and $d\mathrm{PC}(t)/dt > 0$ [30]. If there is a car A that passed RSU_i at physical time t, then A contains $\mathrm{PC}_i(t)$. A vehicle A' from RSU_j arrives RSU_i at time t, and the vehicle's local time is PC_j. The RSU_i sets PC_i to $\max(\mathrm{PC}_i(t), \mathrm{PC}_j + \Delta t)$, where Δt is the traveling delay of a vehicle from RSU_j to RSU_i. In the meantime, the special vehicles record the time difference

$\max(\mathrm{PC}_i(t), \mathrm{PC}_j + \Delta t) - \mathrm{PC}_i(t)$ and historical RSS values, which will be used for data verification and indexing at the central data management server.

29.5 Conclusion

Vehicular data provide a new perspective for many applications in smart city. Unlike the conventional data, where each data is a discrete record, vehicular data is usually a sequence of spatiotemporal records about the surroundings. Compared to location-based services or mobile social networks, the moving trajectories of vehicles within a region are more predicable due to traffic restrictions. Regarding the aspects of security and privacy, this unique feature makes the construction of verifiable vehicular data indexes become cheaper than that of a series of location proofs in a mobile social network, which is strongly dependent on the cryptographic keys among different participants. In this chapter, we propose a new management system by using wireless signal-based environmental data to verify and index vehicular data. Considering that many applications in smart cities are only interested in the correctness of where and when data is collected, in our system, vehicles do not possess any cryptographic key; instead, they simply listen and collect the environmental evidences along their trajectories, and in our system, only the recorded environmental evidences are used to verify/index the vehicular data. In order to guarantee that the collected evidences of different vehicle flows are unique, we deploy several roadside signal transmitters to generate the environmental evidences. Considering the deploying costs and accuracy, we further study the optimal placement problem and time synchronization problem of the transmitters. We believe that the proposed environmental evidence-based vehicular system can bring many new research opportunities to smart cities.

Final Thoughts

In this chapter we discussed the concept of location proof for vehicular trajectory data in smart cities. We overviewed the existing approaches for generating the location proof for a single location spot and provided a set of surrounding environmental information, which can potentially be used to generate the unpredictable, verifiable, and indexable location proofs. We also presented a detailed framework by using the wireless signals from Road Side Units (RSUs) to generate the location proof. And finally we discussed the optimal RSU placement problem and the location–time synchronization problem among RSUs. The concept of environmental evidence-based location proof for vehicular trajectories provides a brand-new research direction in smart cities.

Questions

1 What is the definition of location proof for vehicular trajectory data?

2 How does the conventional location proof disclose user's location privacy?

3 What are the definitions of full distinguishability, full coverage, and secure distinguishability?

4 What is the definition of the optimal RSU placement problem?

5 Why does the synchronization of RSUs' local clocks matter?

References

1 Caragliu, A., Del Bo, C., and Nijkamp, P. (2011) Smart cities in Europe. *Journal of Urban Technology*, **18** (2), 65–82.

2 Lombardi, P., Giordano, S., Farouh, H., and Yousef, W. (2012) Modelling the smart city performance. *Innovation: The European Journal of Social Science Research*, **25** (2), 137–149.

3 Chourabi, H., Nam, T., Walker, S., Gil-Garcia, J.R., Mellouli, S., Nahon, K., Pardo, T., Scholl, H.J. et al., (2012) *Understanding Smart Cities: An Integrative Framework*. IEEE HICSS.

4 Nam, T. and Pardo, T.A. (2011) *Conceptualizing Smart City with Dimensions of Technology, People, and Institutions*. ACM DG.O.

5 Chang, W., Wu, J., and Tan, C.C. (2012) Wormhole defense for cooperative trajectory mapping. *International Journal of Parallel, Emergent and Distributed Systems*, **27** (5), 459–480.

6 Carbunar, B. and Potharaju, R. (2012) *You Unlocked the Mt. Everest Badge on Foursquare! Countering Location Fraud in Geosocial Networks*. MASS, IEEE.

7 He, W., Liu, X., and Ren, M. (2011) *Location Cheating: A Security Challenge to Location-Based Social Network Services*. ICDCS, IEEE.

8 Chen, D., Cho, K.T., Han, S., Jin, Z., and Shin, K.G. (2015) *Invisible Sensing of Vehicle Steering with Smartphones*. ACM MobiSys.

9 Abari, O., Vasisht, D., Katabi, D., and Chandrakasan, A. (2015) Caraoke: an E-toll transponder network for smart cities. *ACM SIGCOMM*, **45** (5), 297–310.

10 Khan, R., Zawoad, S., Haque, M.M., and Hasan, R. (2014) *OTIT: Towards Secure Provenance Modeling for Location Proofs*. ACM ASIACCS.

11 Song, J.H., Wong, V.W., and Leung, V. (2008) *Secure location verification for vehicular Ad-Hoc networks*. IEEE GLOBECOM.

12 Hubaux, J.P., Capkun, S., and Luo, J. (2004) The security and privacy of smart vehicles. *IEEE Security & Privacy Magazine*, **2** (3), 49–55.

13 Abumansoor, O. and Boukerche, A. (2012) A secure cooperative approach for nonline-of-sight location verification in VANET. *IEEE Transactions on Vehicular Technology*, **61** (1), 275–285.

14 Xiao, B., Yu, B., and Gao, C. (2006) *Detection and Localization of Sybil Nodes in VANETs*. ACM DIWANS.

15 Schäfer, M., Lenders, V., and Schmitt, J. (2015) *Secure Track Verification*. IEEE S&P.

16 Zhu, Z. and Cao, G. (2011) *Applaus: A Privacy-Preserving Location Proof Updating System for Location-Based Services*. IEEE INFOCOM.

17 Talasila, M., Curtmola, R., and Borcea, C. (2012) Link: location verification through immediate neighbors knowledge, in *Mobile and Ubiquitous Systems: Computing, Networking, and Services*, Springer.

18 Malandrino, F., Casetti, C., Chiasserini, C.F., Fiore, M., Yokoyama, R.S., and Borgiattino, C. (2013) *A-VIP: Anonymous Verification and Inference of Positions in Vehicular Networks*. IEEE INFOCOM.

19 Zheng, Y., Li, M., Lou, W., and Hou, Y.T. (2012) Sharp: private proximity test and secure handshake with cheat-proof location tags, in *Computer Security–ESORICS 2012*, Springer.

20 Zhang, Y., Tan, C.C., Xu, F., Han, H., and Li, Q. (2015) Vproof: Lightweight privacy-preserving vehicle location proofs. *IEEE Transactions on Vehicular Technology*, **64** (1), 378–385.

21 Higuchi, T., Martin, P., Chakraborty, S., and Srivastava, M. (2015) *Anony-Cast: Privacy-Preserving Location Distribution for Anonymous Crowd Tracking Systems*. ACM UbiComp.

22 Ahmed, S.H., Bouk, S.H., and Kim, D. (2015) RUFS: RobUst forwarder selection in vehicular content-centric networks. *Communications Letters*, **19** (9), 1616–1619.

23 Ahmed, S.H., Bouk, S.H., and Kim, D. (2015) Target RSU selection with low scanning latency in WiMAX-enabled vehicular networks. *Mobile Networks and Applications*, **20** (2), 239–250.

24 Bouk, S.H., Ahmed, S.H., Omoniwa, B., and Kim, D. (2015) Outage minimization using bivious relaying scheme in vehicular delay tolerant networks. *Wireless Personal Communications*, **84** (4), 2679–2692.

25 Sohn, K. and Kim, D. (2008) Dynamic origin–destination flow estimation using cellular communication system. *IEEE Transactions on Vehicular Technology*, **57** (5), 2703–2713.

26 Popa, R.A., Balakrishnan, H., and Blumberg, A.J. (2009) *Vpriv: Protecting Privacy in Location-Based Vehicular Services*. USENIX Security.

27 Lokshantov, D., Vatshelle, M., and Villanger, Y. (2014) *Independent Set in p 5-Free Graphs in Polynomial Time*. ACM-SIAM SODA.

28 Zheng, H., Chang, W., and Wu, J. (2016) *Coverage and Distinguishability Requirements for Traffic Flow Monitoring Systems*. IEEE/ACM IWQoS.

29 Zhou, T., Sharif, H., Hempel, M., Mahasukhon, P., Wang, W., and Ma, T. (2011) A novel adaptive distributed cooperative relaying MAC protocol for vehicular networks. *IEEE Journal on Selected Areas in Communications*, **29** (1), 72–82.

30 Wu, J. (1998) *Distributed System Design*, CRC Press.