sensors

MDPI

Article

Self-Controllable Secure Location Sharing for Trajectory-Based Message Delivery on Cloud-Assisted VANETs

Youngho Park [1], Chul Sur [2], Si-Wan Noh [3] and Kyung-Hyune Rhee [1,*]

[1] Department of IT Convergence and Application Engineering, Pukyong National University, Busan 48513, Korea; pyhoya@pknu.ac.kr
[2] Department of Information Security, Busan University of Foreign Studies, Busan 46234, Korea; certless@gmail.com
[3] Interdisciplinary Program of Information Security, Graduate School, Pukyong National University, Busan 48513, Korea; nosiwan@pukyong.ac.kr
* Correspondence: khrhee@pknu.ac.kr

Received: 30 April 2018; Accepted: 29 June 2018; Published: 1 July 2018

Abstract: In vehicular ad hoc networks, trajectory-based message delivery is a message forwarding strategy that utilizes the vehicle's preferred driving routes information to deliver messages to the moving vehicles with the help of roadside units. For the purpose of supporting trajectory-based message delivery to a moving vehicle, the driving locations of the vehicle need to be shared with message senders. However, from a security perspective, vehicle users do not want their driving locations to be exposed to others except their desired senders for location privacy preservation. Therefore, in this paper, we propose a secure location-sharing system to allow a vehicle user (or driver) to share his/her driving trajectory information with roadside units authorized by the user. To design the proposed system, we put a central service manager which maintains vehicle trajectory data and acts as a broker between vehicles and roadside units to share the trajectory data on the cloud. Nevertheless, we make the trajectory data be hidden from not only unauthorized entities but also the service manager by taking advantage of a proxy re-encryption scheme. Hence, a vehicle can control that only the roadside units designated by the vehicle can access the trajectory data of the vehicle.

Keywords: VANETs; location sharing; authentication; privacy preservation; trajectory-based message delivery

1. Introduction

For the last decade, vehicular ad hoc networks (VANETs) have attracted a great deal of attentions due to the development of vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication technologies. It is a trend of modern vehicles to equip on-board unit (OBU) devices which allow vehicles to communicate with each other as well as roadside units (RSUs) [1]. As a result, up to date, various VANET applications using V2V and V2I communications have been presented to realize not only comfortable driving conditions but also valuable infotainment services on the road. Furthermore, recently, the concept of vehicular networking is extended to vehicular cloud computing by integrating vehicular communications with cloud computing to provide vehicles with pervasive services on the road [2–4].

One of the promising applications is a location-aware service [5] which provides vehicles with useful information for a certain geographic area of interest by taking advantage of vehicular cloud computing. Based on vehicular cloud computing technologies, RSUs deployed on the certain areas can collect and provide local-interesting information to vehicles such as traffic conditions and available

facilities. However, it is challenging on the VANET how we can effectively deliver such service messages to vehicles which are continuously moving on the road. Due to the dynamic mobility and opportunistic connectivity in VANETs, the probability of successful hop-by-hop (among neighboring vehicles) message forwarding to a destination vehicle at a long distance is low and it would result in high message loss. As a solution to this challenge, trajectory-based message delivery with the help of RSUs in VANETs have been researched [6,7].

For example, let us consider a scenario as shown in Figure 1 in which the area around $sRSU_1$ is an interested spot (named socialspot) of a vehicle V_d and $sRSU_1$ provides location-aware service around its local area to V_d. By using trajectory-based message delivery, it would be possible that $sRSU_1$ can efficiently disseminate the service messages to V_d through RSU_1 and RSU_2 acting as relay nodes along V_d's trajectory assuming that $sRSU_1$ knows the driving path of V_d.

On the other hand, driving route or location information of a vehicle is regarded as personal data of the driver, so location privacy is one of the critical security requirements for VANETs as well as identity privacy preservation. In the above trajectory-based message delivery environment, if a vehicle needs to share its driving route with message senders (sRSUs) deployed on some socialspots in which the vehicle is interested for location-aware service, how to share vehicle's trajectory securely with the sRSUs in the system for privacy preservation is required. In other words, the system must be carefully designed not to expose the driving locations of a vehicle to other entities except the sRSUs designated by the vehicle.

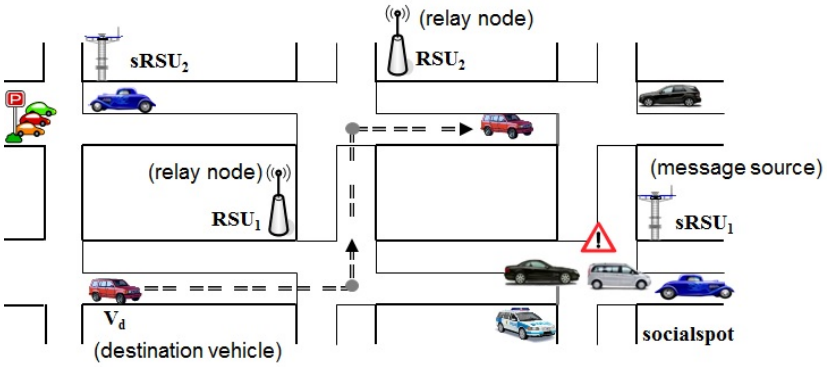

Figure 1. Service scenario of trajectory-based message delivery.

1.1. Related Work

While various VANET applications have been emerging, studies on secure vehicular communications have also been widely performed [4,8–16]. One of the challenging issues is privacy-preserving vehicular communications for protecting location privacy of vehicles in VANETs. To prevent a global eavesdropping attacker from tracking a target vehicle, most privacy-preserving secure vehicular communications recommend using anonymous authentication schemes based on a group signature or pseudonymous credentials of vehicles when exchanging messages [17]. In addition to sender's anonymity, to achieve message receiver's location privacy preservation in VANETs, Lu et al. [18] and Lin et al. [19] proposed secure RSU-assisted message delivery protocols from a source vehicle to a destination vehicle, respectively. However, their work assumes that a receiver vehicle is stationary at a fixed location [18] and the location of a receiver is already known to senders beforehand [19].

For efficient message delivery from an RSU to moving vehicles in VANETS, Jeong et al. proposed an architecture for trajectory-based data delivery [7]. Their work was not focused on security mechanisms for protecting location privacy of vehicles. Instead, they just assume that a control center maintains vehicle trajectories and will not expose the trajectory data to others. In their system,

we cannot help but rely on the control center that the center will carry out its role faithfully. However, from user's viewpoint, the control center may be also a source of concern and users want to control who can access their trajectory data by themselves with a proper security mechanism.

With regard to secure location-sharing in mobile environments, the works in [20–24] proposed some methods to protect user's privacy for location-sharing in mobile online social network services. They are mainly focused on a proximity application to find a nearby friend whose current location is within some distance. Some previous work did not consider preventing a service provider from accessing user's locations data except the work of [20,24]. Even though a service provider is usually involved in location sharing services to distribute the location data of one user to other authorized users, the provider does not need to know the data content. Freudiger et al. [20] and Li et al. [24] proposed a system for protecting user location data from the provider by using data encryption, respectively. However, if a user wants to share his/her location data with multiple friends, the user has to generate multiple encrypted data for the same location data under a different key of each friend or needs interactions to establish a shared secret key with his friends. It burdens the user with computation and communication overheads proportional to the number of friends.

Dong et al. introduced a concept of secure location-sharing based on a proxy re-encryption scheme for mobile applications [25]. Proxy re-encryption is a cryptographic technique to allow a semi-trusted proxy to convert a ciphertext under one party's public key (e.g., V_d in our scenario) into a new ciphertext under another party's public key (e.g., sRSU designated by V_d). While the proxy uses re-encryption keys to perform the conversion, the proxy cannot learn any information about the underlying plaintext. Dong et al. presented an idea that a user can efficiently share its location data with other users as shifting computational overheads for distributing encrypted location data to a proxy, and their idea motivated our work. However, if we adopt an ordinary public key based proxy re-encryption scheme, sRSU has to be involved in generating a re-encryption key to be used for converting a ciphertext by giving its public key certificate to V_d. Moreover, to revoke a re-encryption key of an unwanted sRSU and update re-encryption keys of other valid parties, V_d must change its public key and obtain a new public key certificate. Such interactions are not always available in VANET environments due to the occasional connectivity. Therefore, we need to devise a more flexible and practical method to manage the keys for proxy re-encryption scheme in our VANET service scenario.

In our previous work [26], we presented a secure location sharing based on an id-based proxy re-encryption scheme as considering the non-interactive property of id-based public key cryptography. However, to revoke or update re-encryption keys in the previous work, this system needs to renew user's identity functioning as a public key but it may be impractical to change the identity used in the system. As an alternative, in this paper, we employ a certificateless proxy re-encryption scheme which can provide more flexible key management to our self-controllable secure location-sharing system.

1.2. Our Contribution

Based on the above considerations, in this paper, we propose a secure driving location (trajectory) data sharing system for trajectory-based message delivery in VANETs which can prevent the shared location data on the cloud from being illegitimately exposed to others. We consider a location-aware service scenario in which a vehicle allows some socialspot RSUs designated by the vehicle to access its preferred driving trajectory data stored on the cloud. For secure vehicular communications, group signatures or pseudonymous credentials have been used for anonymous authentication so that the identity and location of a vehicle cannot be linked and tracked on VANETs. However, the main goal of our work is to securely share the trajectory data of a vehicle with only authorized entities from confidentiality view point and to control who can access the location data by vehicle itself. To achieve our goal, we present a system architecture for sharing the trajectory data of vehicles with the help of a service manager acting as a broker between vehicles and socialspot RSUs, then design a secure location-sharing and authenticated message delivery protocols by making use of a certificateless proxy re-encryption scheme and an id-based signature scheme with pseudonymous identities.

In our proposed system, a vehicle can upload driving trajectory data encrypted under its own public key to a semi-trusted service manager on the cloud. The uploaded trajectory data are coupled with re-encryption keys associated with the designated socialspot RSUs so that the manager re-encrypts vehicle's trajectory data and distributes to the intended socialspot RSUs. Then, the socialspot RSUs can send service messages to the vehicle by way of some RSUs along the driving route of the vehicle. We make the vehicle revoke the access rights of unwanted socialspot RSUs by changing vehicle's public key and re-encryption keys but the vehicle does not need to obtain a certificate for the new public key. Therefore, the proposed system can be self-controllable and more flexible.

The rest of this paper is organized as follows: We first present a system architecture and security considerations in Section 2 and cryptographic building blocks to design the proposed system in Section 3. Then, we describe our proposed secure location-sharing and authenticated message delivery protocols in Section 4. We discuss the security and performance of our protocol in Section 5, and finally conclude this paper in Section 6.

2. System Architecture

2.1. Architecture

To design a secure location-sharing system for trajectory-based message delivery on VANETs, we consider the system architecture shown in Figure 2 which consists of trusted authority (TA), service manager (SM), roadside units, and vehicles.

- TA is a fully trusted entity responsible for managing security parameters for the system and issues id-based key pairs to the registered RSUs and vehicles denoted as $\mathcal{R} = \{RSU_1, ..., RSU_m\}$ and $\mathcal{V} = \{V_1, ..., V_n\}$, respectively. TA also manages pseudonymous identities for the vehicles to guarantee anonymity of vehicles on VANET communications.
- SM is a manager which provides storage service on the cloud. To support secure trajectory data sharing, SM maintains encrypted trajectory data of vehicles and acts as a broker which handles to distribute re-encrypted trajectory data to RSUs authorized by the vehicle of the trajectory owner.
- RSUs are subordinated to the TA and sparsely deployed on the roads such as main intersections, and their geographical location information are available through the system. The roles of RSUs in $\mathcal{R}$ are divided into socialspot RSUs and relaying RSUs. Socialspot RSUs (sRSUs), denoted as $\mathcal{SR} = \{sRSU_1, ..., sRSU_l\} \subseteq \mathcal{R}$, are deployed on the specific locations of interest. A set of RSUs establish a local cloud with dedicated servers so that they collect and provide location-aware information to vehicles by means of trajectory-based message delivery. On the other hand, a relaying node RSU equips storage for temporarily holding messages to support message forwarding to the destination vehicles passing by its coverage.
- Vehicles are equipped with OBU and GPS-based navigation system with digital maps. A registering vehicle $V_d \in \mathcal{V}$ selects sRSUs among $\mathcal{SR}$ in which V_d is interested and generates a re-encryption key for the selected sRSUs to share its driving trajectory data through the cloud storage under the control of SM. Whenever V_d changes its preferred driving route, V_d uploads its encrypted driving route data to SM.

Moreover, to clarify the proposed system, we also make the following assumptions.

- Public security parameters of TA are already known to all entities in the system.
- SM and socialspot RSUs are interconnected to each other through a secure and reliable channel.
- Locations and identities of RSUs are publicly available to the system so that vehicles can know which RSUs are deployed at which locations.

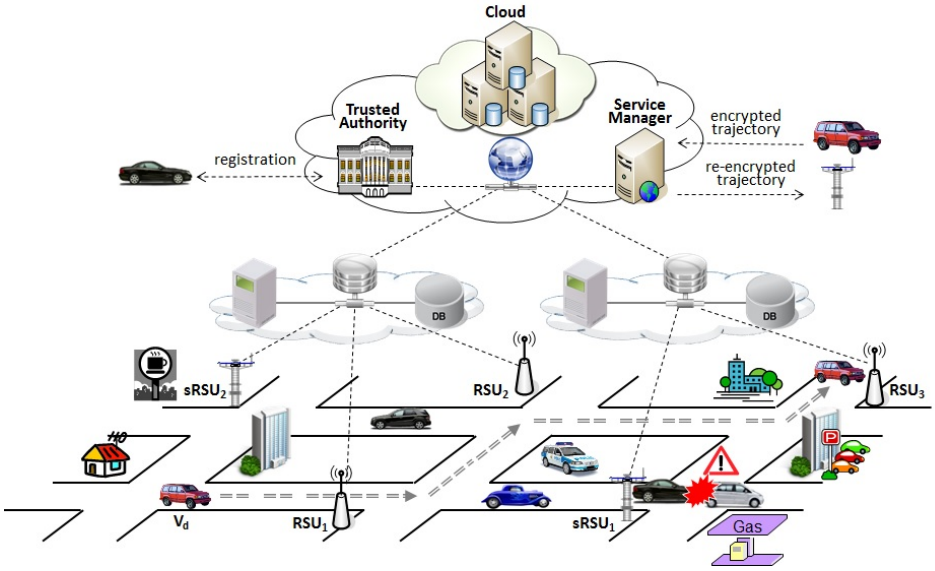

Figure 2. System Architecture.

2.2. Threat Model and Security Considerations

In our system architecture, we consider two types of attack. One is an attack by a compromised SM. In other words, we assume that SM is a honest-but-curious entity that will follow the protocol but may try to extract vehicle's driving trajectory data stored on the cloud for the purpose of collecting and profiling driver's preferences. The other is an attack by an outside attacker which has no access privilege to the cloud but try to learn vehicle's driving locations over VANET communications. However, when we assume that SM and RSUs are interconnected through a secure channel, it is hard for an outsider attacker to control the communications between SM and an RSU. In addition, we assume that all RSUs are trusted under the control of the TA so that SM is not allowed to collude with any RSU. Thus, it is assumed that TA can inspect all RSUs subordinated to it and the collusion or compromise of an RSU is detectable by the TA.

Under the threat model and assumptions, we consider the following security requirements to design a secure vehicle trajectory data sharing system and message delivery protocol to guarantee the location privacy of vehicles in VANETs.

- *Authorized access to trajectory data*: Access to driving trajectory data of a vehicle on the cloud must be restricted to only the RSUs authorized by the owner vehicle of the trajectory data. Even though vehicle's trajectory data are managed under the control of SM, driving trajectory data must be hidden from SM as well as unauthorized entities.
- *Self-control*: When a vehicle uploads its trajectory data to SM, it should be possible for the vehicle to control that which RSUs can or cannot access its trajectory data by the vehicle itself.
- *Authenticated communications*: For secure message delivery on VANETs, vehicles and RSUs involved in message delivery must be authenticated to each others. In message forwarding, a relaying RSU must authenticate a vehicle to check if the vehicle is a valid destination specified in the message. Besides, a vehicle must be convinced that the received message originated from a valid source claimed in the message.
- *Avoiding location tracking*: While a vehicle connects to RSUs on its driving routes to receive messages, driving trajectory of the vehicle must not be tracked by an outside attacker on VANETs. That is, it must be hard for an outside attacker to learn that a vehicle has moved from and to which of RSUs by overhearing vehicle-to-RSU communications.

3. Preliminaries

Before presenting our proposed system, in this section, we briefly outline the properties of a bilinear map and introduce some cryptographic schemes using a bilinear map which serve as the basis of the proposed system. More specifically, we design the proposed system using certificateless proxy re-encryption and id-based signature schemes with pseudonymous identities of vehicles.

3.1. Bilinear Map

Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two cyclic groups of the same prime order q. There is a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfying the following properties.

- Bilinear: $\hat{e}(g^a, g^b) = \hat{e}(g, g)^{ab}$, for all $a, b \in Z_q^*$ and $g \in \mathbb{G}_1$.
- Non-degenerate: If g is a generator of $\mathbb{G}_1$, then $\hat{e}(g, g)$ is a generator of $\mathbb{G}_2$.
- Computable: $\hat{e}(g, h)$ is efficiently computable for any $g, h \in \mathbb{G}_1$.

3.2. Cryptographic Building Blocks

As our building blocks, we adopt a certificateless proxy re-encryption (CL-PRE) scheme [27] and an id-based signature (ID-SIG) scheme [28] described in the followings.

- *Setup*(): A private key generator chooses bilinear map groups $(\mathbb{G}_1, \mathbb{G}_2)$ of the same prime order q, random generators $g, h \in \mathbb{G}_1$, and hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $H_2 : \{0,1\}^* \rightarrow Z_q^*$, $H_3 : G_2 \times G_2 \rightarrow \{0,1\}^{n+\kappa_0}$. It also chooses a random $s \in Z_q^*$ as a master secret then computes $g_1 = g^s$, $g_2 = \hat{e}(g, g)$, $g_3 = \hat{e}(g, h)$. Public system parameters are set as $params := \langle \mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, g, h, g_1, g_2, g_3, H_1, H_2, H_3 \rangle$.
- *idKeyGen*(s, id): Given an identity id, this algorithm outputs an id-based private key $d_{id} = g^{1/(s+H_2(id))}$ under the master secret key s of a private key generator.
- *SetPrvKey*(d_{id}): Given a d_{id} generated by *idKeyGen* for an id, this algorithm chooses random values $a, b \in Z_q^*$ and sets $sk_{id} = (d_{id}, a, b)$ as a private key for CL-PRE for a user of the id.
- *SetPubKey*(sk_{id}): This algorithm returns a public key $pk_{id} = (g_3^a, g^b)$ corresponding to sk_{id} for CL-PRE.
- *clEnc*(m, id, pk_{id}): Certificateless encryption algorithm generates a ciphertext for a given message m under the id and pk_{id} as follows:

  1. Choose a random $\alpha \in \{0,1\}^{\kappa_0}$ for a security parameter κ_0.
  2. Compute $\beta = H_2(m|\alpha|id|pk_{id})$.
  3. Compute $c_1 = (g^{H_2(id)} \cdot g_1)^\beta$, $c_2 = h^\beta$, $c_3 = (m|\alpha) \oplus H_3(g_2^\beta | (g_3^a)^\beta)$, and $c_4 = u^\beta$ where $u = H_1(id|pk_{id}|c_1|c_2|c_3)$.
  4. Output the ciphertext $C = (c_1, c_2, c_3, c_4)$.

- *clReKeyGen*(sk_{id_i}, id_i, id_j, pk_{id_i}, pk_{id_j}): Re-encryption key generation algorithm returns a proxy re-encryption key $rk_{id_i \rightarrow id_j}$ as follows:

  1. Choose a random $\gamma \in Z_q^*$ and compute $\mu = H_2(g_2^\gamma | id_i | pk_{id_i} | id_j | pk_{id_j})$.
  2. Compute $rk^{(1)} = g^{\mu/(s+H_2(id_i))}$, $rk^{(2)} = (g^{H_2(id_j)} \cdot g_1)^\gamma$, and $rk^{(3)} = (g^{b_j})^{a_i}$ where $a_i \in sk_{id_i}$ and $g^{b_j} \in pk_{id_j}$, respectively.
  3. Output the re-encryption key $rk_{id_i \rightarrow id_j} = (rk^{(1)}, rk^{(2)}, rk^{(3)})$.

- *clReEnc*(id_i, pk_{id_i}, C, $rk_{i \rightarrow j}$): Given a ciphertext C under the identity id_i and public key pk_{id_i}, this algorithm outputs a re-encrypted ciphertext C_j delegated to id_j under the $rk_{id_i \rightarrow id_j}$ as follows:

  1. Parse C as $C = (c_1, c_2, c_3, c_4)$

2. Compute $u = H_1(id_i|pk_{id_i}|c_1|c_2|c_3)$ and $y_i = H_2(id_i)$.
3. If $\hat{e}(c_1, u) \stackrel{?}{=} \hat{e}((g^{y_i} \cdot g_1), c_4)$ and $\hat{e}(c_2, u) \stackrel{?}{=} \hat{e}(h, c_4)$ holds,
4. Set the re-encrypted ciphertext $C_j = (c'_1, c'_2, c'_3, c'_4, id_i, pk_{id_i})$, where $c'_1 = \hat{e}(c_1, rk^{(1)})$, $c'_2 = rk^{(2)}$, $c'_3 = \hat{e}(c_2, rk^{(3)})$, and $c'_4 = c_3$.

- *clReDec*(sk_{id_j}, C_j): Given a re-encrypted ciphertext C_j delegated to id_j from id_i, decryption algorithm outputs the message m as follows:

1. Parse C_j as $C_j = (c'_1, c'_2, c'_3, c'_4, id_i, pk_{id_i})$.
2. Compute $\rho = \hat{e}(c'_2, d_{id_j})$ where $d_{id_j} \in sk_{id_j}$, and $\mu = H_2(\rho|id_i|pk_{id_i}|id_j|pk_{id_j})$.
3. Compute $(m|\alpha) = c'_4 \oplus H_3({c'_1}^{1/\mu}|{c'_3}^{1/b_j})$.
4. Compute $\beta = H_2(m|\alpha|id_i|pk_{id_i})$.
5. If $c'_1 \stackrel{?}{=} g_2^{\mu\beta}$ and $c'_3 \stackrel{?}{=} (g_3^{a_i})^{\beta b_j}$ holds, return m. Otherwise output $\perp$.

- *idSig*(d_{id}, m): On input an id-based secret key d_{id} and a message m, this algorithm computes a signature S for the m as follows:

1. Pick a random $x \in Z_q^*$ and compute $\theta = g_2^x$.
2. Set the signature $S = (\sigma_1, \sigma_2)$, where $\sigma_1 = H_2(m, \theta)$ and $\sigma_2 = d_{id}^{x+\sigma_1}$.

- *idVrf*(id, m, S): ID-based signature verification algorithm accepts the message m if $\sigma_1 \stackrel{?}{=} H_2(m, A)$ holds, where $A = \hat{e}(\sigma_2, g^{H_2(id)} g_1) \cdot g_2^{-\sigma_1}$.

4. Proposed System

In this section, we describe the proposed secure trajectory data sharing system for supporting trajectory-based message delivery protocol on VANETs by making use of cryptographic techniques presented in Section 3. The proposed system consists of *setup*, *enrolment*, *trajectory sharing*, and *message delivery* phases. Table 1 shows the notations used to describe the proposed protocol.

Table 1. Notations and descriptions.

Notation	Description
$\mathbb{G}_1, \mathbb{G}_2$	bilinear map groups of a prime order q
$e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$	bilinear map
$g, h \in \mathbb{G}_1$	generators of $\mathbb{G}_1$
$s \in Z_q^*$	master secret key of TA
g^s	public key of TA corresponding to s
RSU_i	identity of a roadside unit
$sRSU_j$	identity of a socialspot RSU
rsk_j	id-based secret key for an RSU_j
pid_{di}	i-th pseudonym of a vehicle V_d
vsk_{di}	id-based private key of a vehicle V_d for pid_{di}
sk_X, pk_X	private and public key of X for CL-PRE
$rk_{V_d \rightarrow sRSU_j}$	re-encryption key of V_d to $sRSU_j$
ts	current timestamp
$Enc_k()$	symmetric encryption under the key k
$Dec_k()$	symmetric decryption under the key k
$MAC_k()$	message authentication code under the key k

4.1. Setup

TA first picks a random $s \in Z_q^*$ as its master secret and runs *Setup*() algorithm to generate public system parameters $params := \langle \mathbb{G}_1, \mathbb{G}_2, q, \hat{e}, g, h, g_1, g_2, g_3, H_1, H_2, H_3 \rangle$. TA also issues id-based

secret keys to RSUs and vehicles registered to the system. Note, at this phase, we assume that id-based secret keys can be issued to RSUs and vehicles through an out-of-band secure channel before they participate in VANETs. Let $\mathcal{R} = \{RSU_1, ..., RSU_m\}$ be the RSUs subordinated by the TA and $\mathcal{V} = \{V_1, ..., V_n\}$ be the registering vehicles. For each $RSU_j \in \mathcal{R}$, TA generates RSU_j's id-based secret key $rsk_j \leftarrow idKeyGen(s, RSU_j)$ and preloads rsk_j to each RSU_j securely. Then, RSU_j sets its private key as $sk_{RSU_j} = (rsk_j, a_j, b_j) \leftarrow SetPrvKey(rsk_j)$ and public key $pk_{RSU_j} = (g_3^{a_j}, g^{b_j}) \leftarrow SetPubKey(sk_{RSU_j})$ for CL-PRE.

On the other hand, for a vehicle $V_d \in \mathcal{V}$, TA chooses $w + 1$ pseudonymous identities $PID_d = \{pid_{di} | 0 \leq i \leq w\}$ for V_d after checking the eligibility of V_d. TA issues V_d's id-based secret keys $vsk_{di} \leftarrow idKeyGen(s, pid_{di})$ for each $pid_{di} \in PID_d$. In our system, pid_{d0} is used for re-encryption procedure while $\{pid_{d1}, ..., pid_{dw}\}$ are used for VANET communications. When obtaining id-based keys for the pseudonyms, V_d sets its private key and public key for proxy re-encryption procedure as $sk_{V_d} = (vsk_{d0}, a_d, b_d) \leftarrow SetPrvKey(vsk_{d0})$ and public key $pk_{V_d} = (g_3^{a_d}, g^{b_d}) \leftarrow SetPubKey(sk_{V_d})$ for CL-PRE. Then, each V_d and each RSU_j, respectively, store (pid_{d0}, pk_{V_d}) and (RSU_j, pk_{RSU_j}) to the cloud storage so as for them to retrieve the public key of each others.

4.2. Enrolment

For a vehicle to receive location-aware service messages by means of trajectory-based message delivery, V_d configures desired socialspots in which V_d is interested for receiving the service messages and selects sRSUs installed at the socialspots denoted as $SR_d = \{sRSU_j | 1 \leq j \leq k\} \subseteq \mathcal{SR}$. Then, to allow each $sRSU_j \in SR_d$ to access V_d's trajectory data under the control of SM using a proxy re-encryption scheme, V_d generates the re-encryption key $rk_{V_d \rightarrow sRSU_j}$ for each $sRSU_j$ as follows:

1. Retrieve $sRSU_j$'s public key pk_{sRSU_j} from the cloud storage.
2. Set a re-encryption key as $rk_{V_d \rightarrow sRSU_j} \leftarrow clReKeyGen(sk_{V_d}, pid_{d0}, sRSU_j, pk_{V_d}, pk_{sRSU_j})$.
3. Compose a message $RSM_d = \{pid_{d0}, pk_{V_d}, SR_d, RK_d\}$, where $RK_d = \{rk_{V_d \rightarrow sRSU_j} \mid sRSU_j \in SR_d\}$.

V_d entrusts RSM_d, consisting of sRSUs list and re-encryption keys, to SM so as for SM to re-encrypt V_d's trajectory data and delegate decryption right to each $sRSU_j$ when V_d uploads its encrypted trajectory data to SM.

4.3. Trajectory Sharing on the Cloud

Let $T_d = \{loc_1 \rightarrow ... \rightarrow loc_t\}$ be the driving trajectory of a vehicle V_d consisting of some specific locations such as main roads or intersections on V_d's preference driving routes. When V_d participates in VANETs, V_d will expects to receive service messages provided by the socialspot RSUs (SR_d) of V_d's interest through some contact point RSUs deployed on V_d's driving trajectory T_d. To securely share V_d's trajectory data and pseudonyms used for message delivery in VANETs with the socialspot RSUs in SR_d through the cloud storage, V_d uploads encrypted trajectory data and pseudonyms to SM as follows:

1. Choose a pseudonym $pid_{di} \in PID_d$ to be used for connecting to a contact point RSU_i.
2. Compose a pseudonym-location pair message $trj_d = \{(pid_{di}, loc_i) | 1 \leq i \leq t\}$.
3. Generate a ciphertext C for trj_d as $C \leftarrow clEnc(trj_d, pid_{d0}, pk_{V_d})$ under V_d's own public key.
4. Upload $TM_d = \{pid_{d0}, C, ts, S_d\}$ to SM, where $S_d \leftarrow idSig(vsk_{d0}, pid_{d0}|C|ts)$ is V_d's signature under the id-based secret key vsk_{d0} of pid_{d0}.

Once V_d uploads its trajectory sharing message TM_d to the cloud storage, SM controls to transform and provide V_d's encrypted trajectory to the sRSUs in SR_d specified by V_d as follows:

1. Parse TM_d as $\{pid_{d0}, C, ts, S_d\}$ and verify the signature S_d as $idVrf(pid_{d0}, pid_{d0}|C|ts, S_d)$ under the given pid_{d0}.

2. Retrieve $RSM_d = \{pid_{d0}, pk_{V_d}, SR_d, RK_d\}$ for the given pid_{d0} from SM's storage.
3. For each $sRSU_j \in SR_d$, extract $rk_{V_d \to sRSU_j} \in RK_d$ and transform the ciphertext C to C'_j as $C'_j \leftarrow clReEnc\ (pid_{d0}, pk_{V_d}, C, rk_{V_d \to sRSU_j})$.
4. Store $\{C'_j, ts\}$ to $sRSU_j$'s directory on the cloud.

While SM maintains vehicles encrypted trajectory information, each socialspot RSU, $sRSU_j$, periodically access its directory to get the updated trajectory of V_d as follows:

1. Downloads $\{C'_j, ts\}$ from its directory on the cloud.
2. Decrypt C'_j to get trj_d as $trj_d = \{(pid_{di}, loc_i) | 1 \leq i \leq t\} \leftarrow clReDec(sk_{sRSU_j}, C'_j)$.
3. Add (pid_{di}, loc_i) pairs to the vehicle list $VList$.

4.4. Trajectory-based Message Delivery

Suppose that $sRSU_j$ collects and provides location-aware service information, such as traffic condition, gas station, and so on. Once $trj_d = \{(pid_{di}, loc_i)\}$ of V_d is loaded, $sRSU_j$ can disseminate location-aware service messages for V_d through an RSU_i deployed on around loc_i. Message delivery from $sRSU_j$ to V_d can be subdivided into three phases: (1) message distribution of $sRSU_j$ to RSU_i; (2) immediate message forwarding by RSU_i to V_d if V_d is within RSU_i's transmission coverage; and (3) message carry-and-forwarding by other vehicles to V_d if V_d is out of RSU_i's transmission coverage, as shown in the Figure 3.

4.4.1. Message Distribution to RSUs

Let msg be a content of location-aware service provided by $sRSU_j$. In this phase, $sRSU_j$ compose a message package M_{jd} for V_d and distributes M_{jd} to V_d's contact point RSU_i as follows:

1. Extract pid_{di} corresponding to loc_i from $VList$.
2. Set a message $M_{jd} = \{pid_{di}, sRSU_j, msg, ts', S_j, ttl\}$ where $S_j \leftarrow idSig(rsk_j, pid_{di} | sRSU_j | msg | ts')$ is $sRSU_j$'s signature and ttl is the message lifetime.
3. Distribute M_{jd} to RSU_i nearby loc_i.

When we assume that RSUs are inter-connected, a message M_{jd} can be easily distributed to other RSUs. During the message delivery, a message bundle has a certain lifetime specified as ttl so that an expired message bundle is discarded. This is for RSUs or carrying vehicles to avoid consuming their storage excessively for a long time even if a target vehicle V_d is not met on the roads.

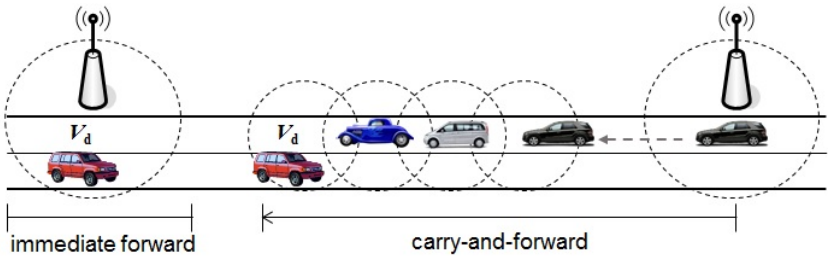

Figure 3. Message forwarding from an RSU to a destination vehicle.

4.4.2. Immediate Message Forwarding

If a message is distributed to contact point RSUs, each RSU stores the received messages and forwards them when a target vehicles for the message passes by RSU's coverage before ttl is expired. Suppose that a vehicle V_d enters RSU_i's coverage and M_{jd} sent from $sRSU_j$ for V_d is kept by RSU_i. The followings describe message forwarding protocol from RSU_i to V_d.

1. RSU_i periodically broadcasts beacon message containing pid_{di} specified as a destination of M_{jd}.
2. If pid_{di} is found in the beacon message, V_d sends a message request $\{req, pid_{di}, S'_d\}$ to RSU_i, where req is a metadata for message requesting and $S'_d \leftarrow idSig(vsk_{di}, req|pid_{di})$.
3. Upon receiving the request message, RSU_i authenticates V_d by verifying S'_d as $idVrf(pid_{di}, req|pid_{di}, S'_d)$. If it holds, RSU_i sends M_{jd} to V_d.

When M_{jd} is downloaded, V_d checks the authenticity of the received message msg to be convinced whether the message actually originated from V_d's desired $sRSU_j$ as follows:

1. Parse $M_{jd} = \{pid_{di}, sRSU_j, msg, ts', S_j, ttl\}$.
2. Verify the signature S_j as $idVrf(sRSU_j, pid_{di}|sRSU_j|msg|ts', S_j)$; and, if it holds,
3. Accept msg as a valid message from $sRSU_j$.

4.4.3. Message Carry-and-Forwarding

If we consider that a target vehicle drives beyond contact point RSU's transmission coverage, another strategy to deliver a message is to rely on VANET routing by means of carry-and-forwarding among neighboring vehicles on the road. For instance, to deliver a message M_{jd} on VANET, RSU_i requests and finds a volunteer vehicle which willingly joins carrying and forwarding the message. Suppose that a vehicle V_c within RSU_i's range is a volunteer vehicle. For RSU_i's request, V_c responds acceptance message $\{acc, pid_{ci}, S_c\}$ to RSU_i to obtain M_{jd}, where acc is a metadata and $S_c \leftarrow idSig(pid_{ci}, acc|pid_{ci})$. RSU_i authenticates V_c by verifying the signature S_c under pid_{ci}, if it holds, provides V_c with M_{jd}.

Once M_{jd} is stored in V_c's storage, V_c carries M_{jd} by itself until V_c runs into the target vehicle V_d of pid_{di} on the road before ttl is expired, or V_c forwards M_{jd} to a next-hop vehicle in accordance with VANET routing protocol. For the former case, if V_c detects V_d on driving, V_c sends notification message to V_d to inform that V_c has a message M_{dj} for V_d. Then, V_d requests the message to V_c in authenticated manner as follows:

1. V_d requests the message to V_c by sending $\{req, pid_{di}, S'_d\}$.
2. V_c verifies the signature S'_d and forwards M_{jd} attached with its signature as $\{res, pid_{ci}, M_{jd}, S_c \leftarrow idSig(vsk_{ci}, res|pid_{ci}|M_{jd})\}$, where res is a metadata for the response.
3. V_d first verifies V_c's signature S_c and extracts M_{jd}. Then, V_d checks if M_{jd} actually originated from $sRSU_j$ by checking $sRSU_j$'s signature S_j in M_{jd} as described in step 2) of immediate forwarding.

On the other hand, if a message is delivered by hop-by-hop forwarding, each intermediary vehicle involved in VANET routing protocol must append its signature to the forwarded message. For instance, suppose that $hop = \{V_1 \rightarrow V_2 \rightarrow ... \rightarrow V_l\}$ is a set of vehicles involved in message forwarding to the destination vehicle V_d (Note, how to establish a route and forward a message to the destination node is beyond our scope. We can assume the existing VANET routing protocols such as [6,29]). The message arriving to V_d consists of $\{fwd, M_{jd}, \{pid_{hi}|S_h\}_{V_h \in hop}\}$ where fwd is a metadata for message forwarding and S_h is a signature of each V_h. The last hop vehicle can forward M_{jd} to V_d as described in the above. By verifying each signature S_h under pid_{hi}, V_d can be convinced that the received message M_{jd} was forwarded via authenticated vehicles.

4.5. Trajectory Update and Sharing Revocation

After vehicles initially upload their trajectory data to SM, they can update their trajectory data whenever they want to change preference driving routes. Remind, in trajectory sharing phase in Section 4.3, V_d's trajectory T_d is uploaded to the cloud in the encrypted form of C in trj_d, and then C is transformed to a re-encrypted ciphertext C_j by the SM to be shared with an $sRSU_j \in SR_d$. Suppose that V_d's trajectory T_d is changed to different driving locations $T'_d = \{loc'_1 \rightarrow ... \rightarrow loc'_t\}$ but V_d still wants to share the changed trajectory with the current sRSUs in SR_d. In this case, the operation required to

V_d is just to generate a ciphertext C' for the changed trajectory T'_d in accordance with the procedures of Section 4.3, and update TM_d to include C' on the cloud storage. Then, SM can transform C' to C'_j for each $sRSU_j$ using the existing re-encryption keys RK_d in RSM_d.

Sometimes, however, a vehicle will not wish to share the updated trajectory data with some sRSUs any more if the vehicle changes its interested socialspots. That is, a vehicle needs to revoke unwilling trajectory sharing. Suppose that V_d does not want to share updated trajectory data with an $sRSU_j$. In the proposed system, V_d can revoke trajectory sharing with $sRSU_j$ by updating V_d's private and public key pair and re-encryption keys for sRSUs except the $sRSU_j$ as follows:

1. Renew its private key as $sk_{V_d} = (vsk_{d0}, a'_d, b'_d) \leftarrow SetPrvKey(vsk_{d0})$ and public key as $pk_{V_d} = (g_3^{a'_d}, g^{b'_d}) \leftarrow SetPubKey(sk_{V_d})$ by choosing new random values $a'_d, b'_d \in Z_q^*$.
2. Change the socialspot RSUs list as SR_d to SR'_d by adding new sRSUs and deleting revoked sRSUs.
3. Set $RK'_d = \{rk_{V_d \rightarrow sRSU_i} \mid sRSU_i \in SR'_d\}$ by running $rk_{V_d \rightarrow sRSU_i} \leftarrow clReKeyGen(sk_{V_d}, pid_{d0}, sRSU_i, pk_{V_d}, pk_{sRSU_i})$ for each $sRSU_i$.
4. Replace the existing RSM_d uploaded in enrolment phase of Section 4.2 with the updated RSM'_d including RK'_d and SR'_d.

5. Analysis

5.1. Security

5.1.1. Authorized Access to Trajectory Data

Since the locations of a vehicle can be regarded as personal information of a driver, driving trajectory data maintained on the cloud must not be exposed to not only outsiders but also SM illegally in the system. In our proposed system, a trajectory data trj_d consisting of pseudonym-location pairs of a vehicle V_d is maintained by SM in the encrypted form under V_d's public key pk_{V_d} as $C \leftarrow clEnc(trj_d, pid_{d0}, pk_{V_d})$. Essentially, nobody can gain V_d's trajectory data as trying to decrypt C without knowing V_d's private key sk_{V_d} corresponding to pk_{V_d}. On the other hand, some sRSUs of V_d's interested socialspots specified in SR_d need to know V_d's driving trajectories for disseminating service messages through contact point RSUs along V_d's driving routes. V_d can allow an $sRSU_j \in SR_d$ to get its trajectory data by giving a re-encryption key $rk_{V_d \rightarrow sRSU_j} \leftarrow clReKeyGen(sk_{V_d}, pid_{d0}, sRSU_j, pk_{V_d}, pk_{sRSU_j})$ to SM instead of entrusting plaintext trajectory data. The encrypted trajectory data C are then re-encrypted to $C'_j \leftarrow clReEnc\ (pid_{d0}, pk_{V_d}, C, rk_{V_d \rightarrow sRSU_j})$ for $sRSU_j$ under the key $rk_{V_d \rightarrow sRSU_j}$ by SM. Hence, only a valid $sRSU_j$ that possesses a private key sk_{sRSU_j} corresponding to pk_{sRSU_j} involved in $rk_{V_d \rightarrow sRSU_j}$ can get trj_d by decrypting C_j.

In addition to an outside attacker and unauthorized RSUs, another concern is the security threat of SM to the trajectory data managed on the cloud since a semi-honest SM may be curious to know vehicle's driving locations for the purpose of collecting and profiling driver's preferences. At this phase, even though the encrypted trajectory data and re-encryption keys are given to SM, it is hard for SM to deduce the private key of V_d or $sRSU_j$ for the purpose of recovering the trajectory data trj_d from C or C_j if we assume the security properties of the underlying proxy re-encryption scheme [27]. Therefore, neither SM nor unauthorized RSUs can access vehicle's trajectory data on the cloud unless the vehicle authorizes decryption rights for the encrypted trajectory data under re-encryption keys.

5.1.2. Self-Controllable Trajectory Sharing

In our proposed system, even though vehicle's shared trajectory data are maintained and transferred by SM, it is the owner vehicle of the trajectory data that can decide what RSUs can access its trajectory data on the cloud. As mentioned before, distribution of the trajectory data of V_d is performed by SM according to socialspot list SR_d and re-encryption keys $RK_d = \{rk_{V_d \rightarrow sRSU_j} | sRSU_j \in SR_d\}$ generated by V_d in the enrolment phase. If V_d does not want an $sRSU_j$ existing in SR_d to access its

trajectory data any more, V_d can revoke $sRSU_j$'s decryption right by generating new re-encryption keys for sRSUs in SR_d except $sRSU_j$. Once V_d uploads updated $SR'_d \setminus \{sRSU_j\}$ and RK'_d for each sRSU in SR'_d, SM will exclude $sRSU_j$ and not provide $sRSU_j$ with the re-encrypted trajectory data of V_d. In our revocation procedure of Section 4.5, V_d can generate re-encryption keys without involvement of SM and sRSUs, and it does not require renewing public keys of sRSUs due to the functionality of certificateless proxy re-encryption scheme. Therefore, our proposed system can provide a flexible and self-controllable trajectory data sharing mechanism.

5.1.3. Authenticated Vehicular Communications

To receive a service message in the form of $M = \{pid_{di}, sRSU_j, msg, ts', S_j, ttl\}$ served by sRSUs on VANETs, V_d must be authenticated to a contact point RSU_i or carrier vehicle V_c as V_d is the valid destination vehicle of pid_{di} specified in the message M. In message forwarding phase of Section 4.4, when V_d requests a message M kept by a contact point RSU_i or a carrying vehicle V_c, V_d must present its id-based signature $S'_d \leftarrow idSig(vsk_{di}, req|pid_{di})$ which is in turn verified under the given pid_{di} of the message M. If we assume the security of an id-based signature scheme, any vehicle which does not know the private key corresponding to pid_{di} except V_d cannot forge the signature nor impersonate V_d. Therefore, only the valid vehicle V_d authenticated under the pid_{di} can receive the message M.

In addition, when V_d receives a message M, V_d also authenticates the message sender sRSU by verifying the attached id-based signature S_j under the sRUS's identity $sRSU_j$ specified in M. That is, V_d can be convinced that the message M was sent from the $sRSU_j$, in which V_d is interested, if the signature S_j is verified as valid under the id of $sRSU_j$.

5.1.4. Avoiding Location Tracking

Due to the access control to the trajectory data, we can prevent an outside attacker as well as any unauthorized entity from learning vehicle's driving trajectory stored on the cloud. On the other hand, to prevent an outside attacker from tracking driving path of a vehicle by eavesdropping on the vehicle-to-RSU communications, it must be difficult for an outside attacker to guess that the observed vehicle at different RSU's coverage is the same vehicle while a vehicle connects to contact point RSUs for receiving a message over VANET on its driving. In our proposed system, a vehicle V_d has a set of pseudonyms $\{pid_{d1}, .., pid_{dw}\}$, which can be independently generated random values, and it is recommended to use a different pseudonym for identification and authentication whenever V_d connects to a different contact point RSU_i deployed at the location of loc_i along V_d's driving path. Therefore, we can make it hard for an outside attacker to track moving locations of a vehicle if any two pseudonyms $pid_{di} \neq pid_{dj}$, respectively, observed by the attacker at RSU_i and RSU_j are unlinkable to the same vehicle from attacker's viewpoint.

Moreover, $\langle pid_{di}, loc_i \rangle$ pairs are only known to the sRSUs authorized by V_d by means of a re-encryption technique. Another possible attack for an outside attacker is to compromise an sRSU to extract V_d's pseudonyms and driving trajectory data $\langle pid_{di}, loc_i \rangle$ kept by the sRSU on the VANET. As a countermeasure to this threat, in this paper, we just assume that all RSUs are inspected by the TA and compromise of an RSU can be preventable and detectable by means of a security module such as tamper proof device. RSUs would not disclose any information without the authorization of the TA. Nevertheless, if an sRSU is detected as abused, TA can take an action to recover the sRSU and the vehicle can generate new re-encryption keys for sRSUs to protect the changed trajectory data after that.

5.2. Performance

We simulated the message delivery on VANETs to evaluate the impact of the proposed security protocol to the performance of message forwarding from RSUs to destination vehicles. We implemented the simulation by using NS-2 and SUMO [30] simulators as considering an urban road environment. Figure 4 and Table 2 show the 4600 m $\times$ 3800 m road configuration and simulation settings, respectively.

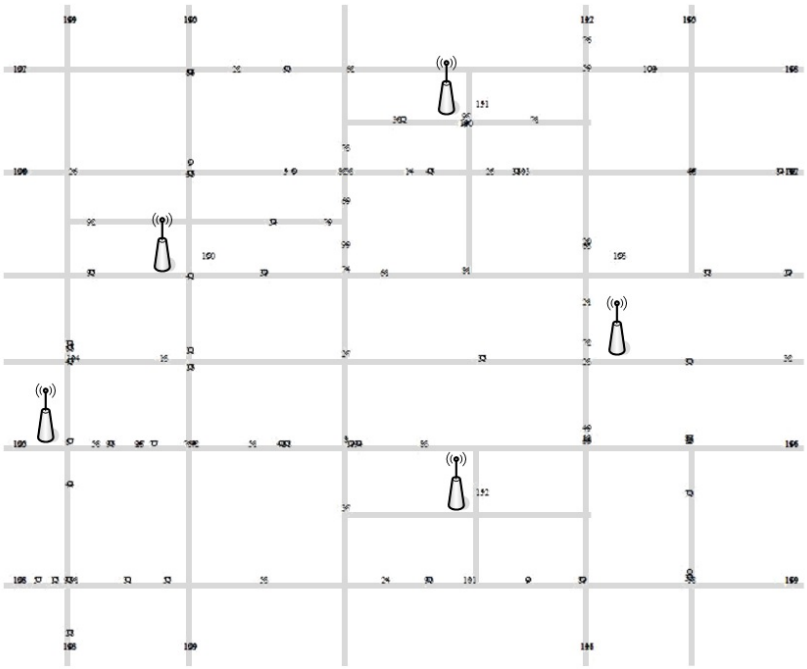

Figure 4. Road configuration for simulation.

Table 2. NS-2 simulation parameters.

Simulation Setting	
road dimension	4600 m × 3800 m
# of vehicles	{30, 45, 60, 75, 90, 105, 120, 135, 150}
# of contact point RSUs	5
# of destination vehicles	15
moving speed	{40, 50, 60, 70} km/h
mobility model	Manhattan model
wireless/bandwidth	802.11 p/6 Mbps
radio coverage	250 m
message size	100 KB
message lifetime	500 s
simulation time	2000 s

For our simulation, we applied Manhattan mobility model in which each vehicle moves in horizontal or vertical direction on an urban road and the probability of going straight is 0.5 and taking a left or right is, respectively, 0.25 [31]. We varied the number of vehicles from 30 to 150 moving with 11.1 m/s (40 km/h) to 19.4 m/s (70 km/h) speed on average, and put five contact point RSUs relaying messages to 15 destination vehicles. For message carry-and-forwarding, we adopted the DTN routing protocol of [32] and adjusted message forwarding time to compensate for the delay caused by the authentication process. To measure the authentication overhead of message delivery, we used the benchmark results of pairing-based cryptography library [33] implemented on Intel Quad Core2 2.4 GHz machine by using the supersingular curve $y^2 = x^3 + x$ for the group $\mathbb{G}_1$ with 512-bit base field size and 160-bit group order providing 1024-bit security. Then, we evaluated the performance in terms of message delivery delay and successful delivery ratio on average of 15 destination vehicles for each experiment by varying the number of vehicles and their moving speed.

Let D be the total message delay from a contact point RSU to a destination vehicle. The message delay can be estimated as $D = D_{tr} + D_{auth}$ where D_{tr} is the delay for message transmission and D_{auth}

is for authentication process resulting from signature generation and verification. Depending on message forwarding method, D_{auth} can be classified as

$$D_{auth} = \begin{cases} T_{sig} + T_{vrf}, & \text{immediate;} \\ 2(T_{sig} + T_{vrf}) + \sum_{i=1}^{l-1} T_{sig_i}, & \text{carry-forward.} \end{cases}$$

where l is the number of hops and T_{sig} and T_{vrf} are the times for signature generation and verification evaluated as T_{sig} = 4.65 ms and T_{vrf} = 10.93 ms, respectively. For immediate forwarding, it requires signature generation of the destination vehicle and verification of a contact point RSU before message forwarding. On the other hand, for carry-and-forwarding, authentication of the first carrier vehicle to a contact point RSU and authentication between the last hop vehicle and the destination vehicle are required while each intermediary vehicle only appends its signature to the forwarded message.

With regard to our experiments depending on the number of vehicles and the moving speed, the initial positions of vehicles are randomly generated and the distributions of vehicles on the road in each experiment are not consistent. Thus, those experiments are independent of each other and cause uneven variations in the results of Figures 5–8. However, we would like to estimate the performance by observing the trend of changes through the experiment results.

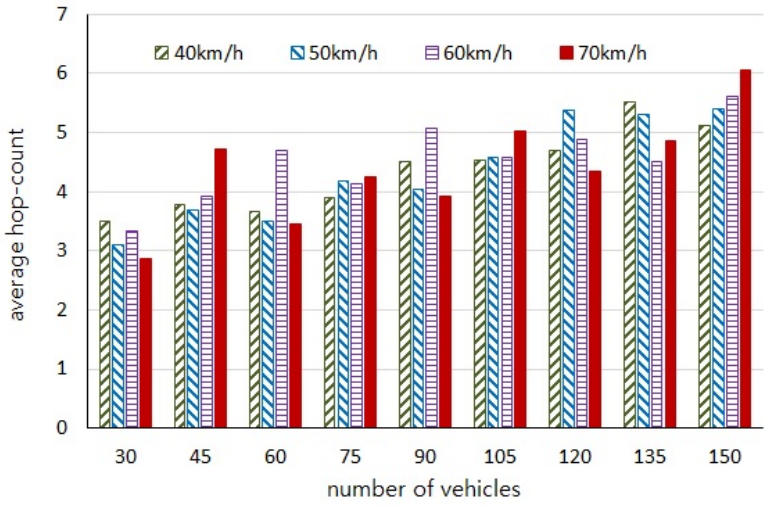

Figure 5. Average number of hops for forwarding messages to the destination vehicles.

Figure 5 shows the average number of hops, and Figure 6 shows the average authentication delay evaluated by our simulations, respectively. We can observe that the number of intermediary vehicles participating in message forwarding is increased as the more vehicles are distributed on the road. It is obvious that the authentication delay gets longer as the number of hops are increased. However, it should be noted that the more vehicles participate in message forwarding the shorter message delay occurs, as shown in Figure 7, because the carry delay depending on the moving speed of vehicles is much longer than the communication delay. Therefore, the authentication delay is insignificant and has little effect on the total message delay.

We also evaluated the successful message delivery ratio for 500 s of message lifetime and Figure 8 shows the results. As aforementioned, if more vehicles are distributed on the VANET and move with high speed, vehicles can have more chance to meet other vehicles which results in higher possibility of message carrying-and-forwarding as well as shorter message delay. From our simulation results, we can see that almost all of the messages can be successfully delivered to the destination vehicles within 350 s when we put more than 135 vehicles moving with higher than 50 km/h speed.

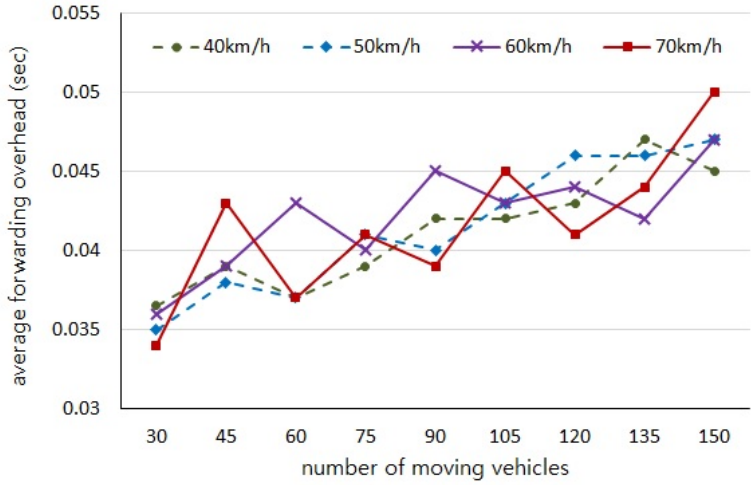

Figure 6. Average forwarding delay (D_{auth}) to the number of hops burdened by authentication process.

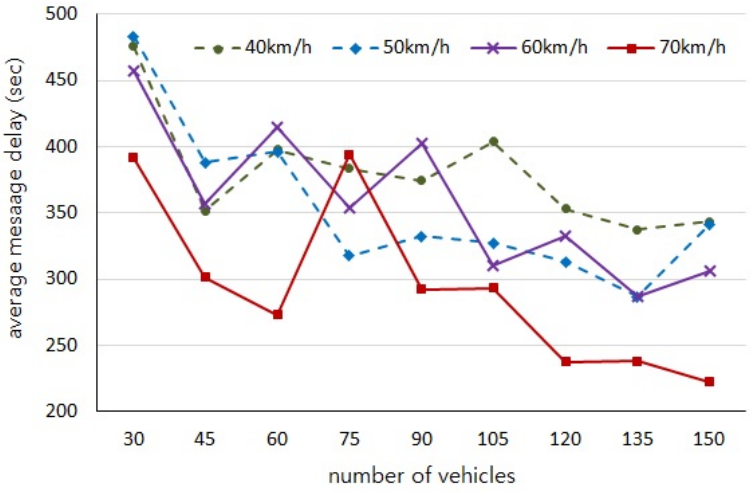

Figure 7. Average message delivery delay (D) from contact point RSUs to the destination vehicles.

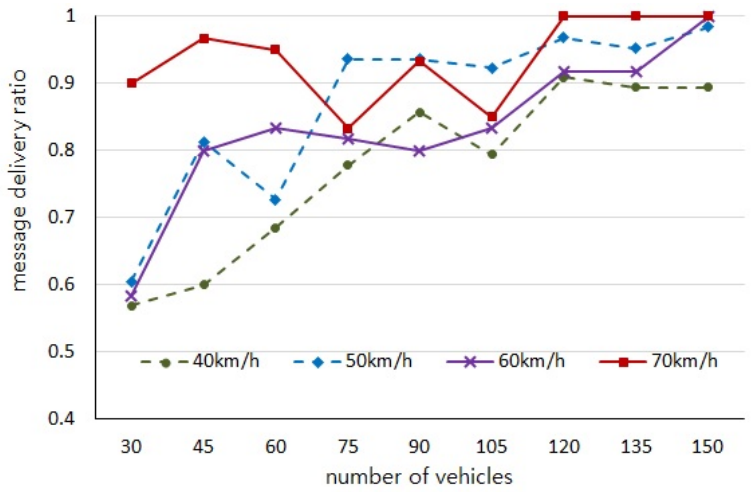

Figure 8. Successful message delivery ratio for the destination vehicles.

6. Conclusions

Trajectory-based message delivery with the help of roadside units have been considered as an efficient message dissemination method on VANETs under the assumption that message senders know the driving routes of message receiver vehicles. However, from a security viewpoint of location privacy, users of VANETs want to limit sharing of their driving locations to the desired message senders by themselves to prevent the location information from being illegitimately exposed to others. Therefore, in this paper, we propose a secure location sharing system in which vehicles control what roadside units can access vehicles driving locations on their own decision for trajectory-based message delivery services. To effectively share vehicle's trajectory data with the socialspot roadside units designated by the vehicle on the cloud, we devised a secure trajectory data sharing mechanism by taking advantage of a certificateless proxy re-encryption scheme in which the role of maintaining and distributing encrypted trajectory data can be delegated to a semi-trusted service manager but the access rights to the trajectory data are controlled by vehicles themselves. Therefore, even though vehicles trajectory data are managed by a service manager on the cloud, the trajectory data are hidden from not only unauthorized entities but also the service manager. In addition, whenever vehicles change their preferred driving routes, vehicles can efficiently revoke the access rights of unwanted roadside units to the updated trajectory data just by updating re-encryption keys without involvement of the service manager and roadside units. Consequently, we can design a more flexible and self-controllable secure trajectory data sharing system on VANETs.

Author Contributions: Y.P. and C.S. have been involved in all stages of this work including protocol design, validation, and writing & editing the manuscript; S.-W.N. performed simulation and experiment analysis; K.-H.R. supervised the work.

Funding: This work was partially supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2018-2015-0-00403) supervised by the IITP (Institute for Information & communications Technology Promotion), and by the IITP grant funded by the Korea government (MSIT) (No. 2017-0-00156, The Development of a Secure Framework and Evaluation Method for Blockchain).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kenney, J.B. Dedicated Short-Range Communications (DSRC) Standards in the United States. *Proc. IEEE* **2011**, *99*, 1162–1182. [CrossRef]
2. Olariu, S.; Hristov, T.; Yan, G. The next paradigm shift: From vehicular networks to vehicular clouds. In *The Cutting Edge Directions, Mobile Ad Hoc Networking*; Wiley: Hoboken, NJ, USA, 2013; pp. 645–700.
3. Yu, R.; Zhang, Y.; Gjessing, S.; Xia, W.; Yang, K. Toward cloud-based vehicular networks with efficient resource management. *IEEE Netw.* **2013**, *27*, 49–55. [CrossRef]
4. Zhang, L.; Men, X.; Choo, K.K.R.; Zhang, Y.; Dai, F. Privacy-preserving cloud establishment and data dissemination scheme for vehicular cloud. *IEEE Trans. Dependable Secure Comput.* **2018**. [CrossRef]
5. Dikaiakos, M.D.; Florides, A.; Nadeem, T.; Iftode, L. Location-aware services over vehicular ad-hoc networks using car-to-car communication. *IEEE J. Sel. Areas Commun.* **2007**, *25*, 1590–1602. [CrossRef]
6. Jeong, J.; Guo, S.; Gu, Y.; He, T.; Du, D.H. Trajectory-based data forwarding for light-traffic vehicular ad hoc networks. *IEEE Trans. Parallel Distrib. Syst.* **2011**, *22*, 743–757. [CrossRef]
7. Jeong, J.; Guo, S.; Gu, Y.; He, T.; Du, D.H. Trajectory-based statistical forwarding for multihop infrastructure-to-vehicle data delivery. *IEEE Trans. Mob. Comput.* **2012**, *11*, 1523–1537. [CrossRef]
8. Raya, M.; Hubaux, J.P. Securing vehicular ad hoc networks. *J. Comput. Secur.* **2007**, *15*, 39–68. [CrossRef]
9. Calandriello, G.; Papadimitratos, P.; Hubaux, J.P.; Lioy, A. Efficient and robust pseudonymous authentication in VANET. In Proceedings of the 4th ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2007), Montreal, QC, Canada, 10 September 2007; pp. 19–28.
10. Lu, R.; Lin, X.; Zhu, H.; Ho, P.H.; Shen, X. ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. In Proceedings of the 27th Conference on Computer Communications, IEEE INFOCOM'08, Phoenix, AZ, USA, 13–18 April 2008; pp. 1229–1237.

11. Jung, C.D.; Sur, C.; Park, Y.; Rhee, K.H. A robust and efficient anonymous authentication protocol in VANETs. *J. Commun. Netw.* **2009**, *11*, 607–614. [CrossRef]
12. Park, Y.; Sur, C.; Jung, C.D.; Rhee, K.H. An efficient anonymous authentication protocol for secure vehicular communications. *J. Inf. Sci. Eng.* **2010**, *26*, 785–800.
13. Zhang, L.; Wu, Q.; Solanas, A.; Domingo-Ferrer, J. A scalable robust authentication protocol for secure Vehicular communications. *IEEE Trans. Veh. Technol.* **2010**, *59*, 1606–1617. [CrossRef]
14. Park, Y.; Sur, C.; Rhee, K.H. Pseudonymous authentication for secure V2I services in cloud-based vehicular networks. *J. Ambient Intell. Humaniz. Comput.* **2016**, *7*, 661–671. [CrossRef]
15. Zhang, L.; Hu, C.; Wu, Q.; Domingo-Ferrer, J.; Qin, B. Privacy-preserving vehicular communication authentication with hierarchical aggregation and fast response. *IEEE Trans. Comput.* **2016**, *65*, 2562–2574. [CrossRef]
16. Zhang, L.; Wu, Q.; Domingo-Ferrer, J.; Qin, B.; Hu, C. Distributed aggregate privacy-preserving authentication in VANETs. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 516–526. [CrossRef]
17. Lu, H.; Li, J. Privacy-preserving authentication schemes for vehicular ad hoc networks: A survey. *Wirel. Commun. Mob. Comput.* **2016**, *16*, 643–655. [CrossRef]
18. Lu, R.; Lin, X.; Shen, X. SPRING: A social-based privacy-preserving packet forwarding protocol for vehicular delay tolerant networks. In Proceedings of the 2010 IEEE INFOCOM'10, San Diego, CA, USA, 14–19 March 2010; pp. 632–640.
19. Lin, X.; Lu, R.; Liang, X.; Shen, X. STAP: A social-tier-assisted packet forwarding protocol for achieving receiver-location privacy preservation in VANETs. In Proceedings of the IEEE INFOCOM'11, Shanghai, China, 11–15 April 2011; pp. 2147–2155.
20. Freudiger, J.; Neu, R.; Hubaux, J.P. Private sharing of user location over online social networks. In Proceedings of the 3rd Hot Topics in Privacy Enhancing Technologies (HotPETs'10), Berlin, Germany, 21–23 July 2010.
21. Wei, W.; Xu, F.; Li, Q. MobiShare: Flexible privacy-preserving location sharing in mobile online social networks. In Proceedings of the IEEE INFOCOM'12, Orlando, FL, USA, 25–30 March 2012; pp. 2616–2620.
22. Li, J.; Li, J.; Chen, X.; Liu, Z.; Jia, C. MobiShare+: Security Improved System for Location Sharing in Mobile Online Social Networks. *J. Internet Serv. Inf. Secur.* **2014**, *4*, 25–36.
23. Liu, Z.; Luo, D.; Li, J.; Chen, X.; Jia, C. N-Mobishare: new privacy-preserving location-sharing system for mobile online social networks. *Int. J. Comput. Math.* **2016**, *93*, 384–400. [CrossRef]
24. Li, J.; Yan, H.; Liu, Z.; Chen, X.; Huang, X.; Wong, D.S. Location-sharing systems with enhanced privacy in mobile online social networks. *IEEE Syst. J.* **2017**, *11*, 439–448. [CrossRef]
25. Dong, C.; Dulay, N. Longitude: A privacy-preserving location sharing protocol for mobile applications. In Proceedings of the IFIP International Conference on Trust Management, IFIPTM 2011, IFIPAICT, Copenhagen, Denmark, 29 June–1 July 2011; Springer: Berlin, Germany, 2011; Volume 358, pp. 133–148.
26. Park, Y.; Sur, C.; Noh, S.W.; Rhee, K.H. Secure vehicle location-sharing for trajectory-based message delivery on VANETs. In Proceedings of the IEEE 26th International Symposium on Industrial Electronics (ISIE'17), Edinburgh, UK, 19–21 June 2017; pp. 1451–1456.
27. Sur, C.; Jung, C.D.; Park, Y.; Rhee, K.H. Chosen-ciphertext secure certificateless proxy re-encryption. In *Communication and Multimedia Security—CMS 2010*; Springer: Berlin, Germany, 2010; Volume 6109, pp. 214–232.
28. Barreto, P.S.L.M.; Libert, B.; McCullagh, N.; Quisquater, J.J. Efficient and provably-secure identity-baed signatures and signcryption from bilinear maps. In *Advances in Cryptology—ASIACRYPT 2005*; Springer: Berlin, Germany, 2005; Volume 3788, pp. 515–532.
29. Zhao, J.; Cao, G. VADD: Vehicle-assisted data delivery in vehicular ad hoc networks. *IEEE Trans. Veh. Technol.* **2008**, *57*, 1910–1922. [CrossRef]
30. Simulation of Urban MObility. Available online: http://sumo.dlr.de (accessed on 30 June 2018).
31. Bai, F.; Sadagopan, N.; Helmy, A. IMPORTANT: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks. In Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM'13), San Francisco, CA, USA, 30 March–3 April 2013; pp. 825–835.

32. Lakkakorpi, J.; Pitkanen, M.; Ott, J. Adaptive routing in mobile opportunistic networks. In Proceedings of the 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems, Bodrum, Turkey, 17–21 October 2010; pp. 101–109.
33. Pairing-based Cryptography Library. Available online: https://crypto.stanford.edu/pbc/ (accessed on 30 June 2018).