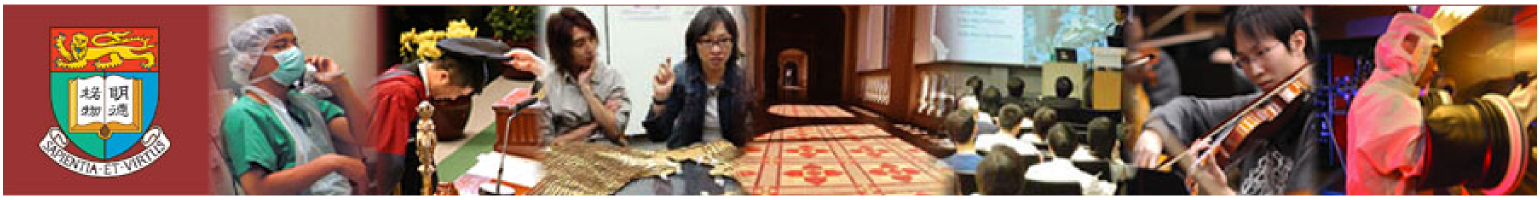

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

VSPN: VANET-based Secure and Privacy-preserving Navigation

T.W. Chim, S.M. Yiu and Lucas C.K. Hui
Department of Computer Science
The University of Hong Kong
Email: {twchim, smyiu, hui}@cs.hku.hk

Victor O.K. Li
Department of Electrical and Electronic Engineering
The University of Hong Kong
Email: vli@eee.hku.hk

***Abstract*—In this paper, we propose a navigation scheme that utilizes the online road information collected by a vehicular ad hoc network (VANET) to guide the drivers to desired destinations in a real-time and distributed manner. The proposed scheme has the advantage of using real-time road conditions to compute a better route and at the same time, the information source can be properly authenticated. To protect the privacy of the drivers, the query (destination) and the driver who issues the query are guaranteed to be unlinkable to any party including the trusted authority. We make use of the idea of anonymous credential to achieve this goal. In addition to authentication and privacy-preserving, our scheme fulfills all other necessary security requirements. Using the real maps of New York and California, we conducted a simulation study on our scheme showing that it is effective in terms of processing delay and providing routes of much shorter travelling time.**

***Index Terms*—Navigation, secure vehicular sensor network, signature verification, pseudo identity, anonymous credential, proxy re-encryption**

I. Introduction

Finding a route to a certain destination is a common experience for all drivers. In the old days, a driver usually refers to a hardcopy of the atlas. The drawbacks are quite obvious. With the introduction of Global Positioning System (GPS) [1], GPS-based navigation systems become popular. [2] is an example. In such a system, a small hardware device is installed on a vehicle. By receiving GPS signals, the device can determine its current location and then find the geographically shortest route to a certain destination based on a local map database. However, the route searching procedure of these systems is based on a local map database and real-time road conditions are not taken into account.

To learn about real-time road conditions, a driver needs another system known as Traffic Message Channel (TMC) [3] which has been adopted in a number of developed countries. TMC makes use of FM radio data system to broadcast real-time traffic and weather information to drivers. A special equipment is required to decode or to filter the information received. However, only special road conditions (e.g. severe taffic accident) are broadcasted and a driver cannot obtain information like the general fluency of a road from TMC.

Recently, vehicular ad hoc network (VANET) becomes increasingly popular in many countries. It is an important element of the Intelligent Transportation Systems (ITSs) [4]. In a typical VANET, each vehicle is assumed to have an on-board unit (OBU) and there are road-side units (RSU) installed along the roads. A trusted authority (TA) and maybe some other application servers are installed in the backend. The OBUs and RSUs communicate using the Dedicated Short Range Communications (DSRC) protocol [5] over the wireless channel while the RSUs, TA, and the application servers communicate using a secure fixed network (e.g. the Internet). The basic application of a VANET is to allow arbitrary vehicles to broadcast safety messages (e.g. vehicle speed, turning direction, traffic accident information) to other nearby vehicles (denoted as vehicle-vehicle or V2V communications) and to RSU (denoted as vehicle-infrastructure or V2I communications) regularly such that other vehicles may adjust their travelling routes and RSUs may inform the traffic control center to adjust traffic lights for avoiding possible traffic congestion. As such, a VANET can also be interpreted as a sensor network because the traffic control center or some other central servers can collect lots of useful information about road conditions from vehicles. It is natural to investigate how to utilize the collected real-time road conditions to provide useful applications.

In this paper, we propose a new application - VANET-based Secure and Privacy-preserving Navigation (VSPN) which makes use of the collected data to provide navigation service to drivers. Based on the destination and the current location of the driver (the query), the system can automatically search for a route which yields minimum travelling delay in a distributed manner using the online information of the road condition. In addition of driving guidance, the navigation results can also be used for other purposes. For example, a recent work [6] proposed to use the returned routes for opportunistically routing multimedia information such as images and videos about a desired scene to vehicles.

Like other communication networks, security issues have been widely addressed in VANETs. Any navigation scheme must also satisfy these security requirements. First, whether or not the service is free, subscription to the service is usually required. A user (note that since the user is usually the driver and it is associated with the vehicle, we use these terms interchangeably throughout the paper) must be authenticated to ensure he is a valid subscriber. Messages sent in the system must be authenticated and signed to make sure that they were not modified by anyone.

On the other hand, since a vehicle's OBU will continuously

Digital Object Indentifier 10.1109/TC.2012.188

communicate with RSUs, the driving habit of a driver as well as the travelling routes can be easily analyzed. So, privacy protection is another basic requirement in VANETs. One common approach to resolve this possible privacy leakage is to use a different authenticable, but unrelated pseudo identity to communicate with a different RSU. Thus, collecting all messages between a vehicle and all RSUs cannot link the messages together to reconstruct the driving routes or analyze the driving habit of a driver. However, to protect the system, for example, if a subscriber launches a denial-of-service attack to the system by sending out many messages to an RSU in a short period of time, the system administrator should be able to trace that user and to block its further access to protect the system. Therefore, though pseudo identity is used, a trusted party (e.g. TA) should be able to obtain a user's real identity if necessary. In other words, the real identity of a vehicle should be kept anonymous from other vehicles as well as RSUs and anyone (including all RSUs) cannot reveal the real identify of a vehicle by analyzing multiple messages sent by it. But the authorized party (TA) is able to retrieve the real identity of the sender if needed based on its pseudo identity.

For a VANET-based navigation system, we need additional security and privacy requirements which make the problem non-trivial. In a basic VANET system, the trusted authority has the power to reveal the real identity of a vehicle. If the navigation system is not carefully designed, it means that the real identity of a driver and the query issued by him can be easily linked up and analyzed. While we still want the TA to have the authority to reveal the real identity based on a pseudo identity, we want to ensure that the TA does not know where the driver wants to go.

Basic confidentiality is another important factor in our scheme. First, a driver may not want vehicles nearby to know his/her destination by eavesdropping his/her query issued. Second, when the system sends the navigation result back to him/her, we do not want non-subscribers nearby to enjoy free navigation service in case it is going to the same destination, in order to protect the profit of the operator if the service is a charged item. Moreover, since navigation involves the information (including locations and road conditions) provided by more than one RSU and RSUs are left unattended at roadsides most of the time, proper authentication of this information becomes critical. Moreover, the authentication must be efficient, otherwise, the querying duration will be unacceptably long.

To summarize, our VSPN scheme adopts some security primitives in a non-trivial way to provide the following security features: 1) When using the navigation service, a vehicle can be properly authenticated. Privacy is preserved using the idea of pseudo identity. At the same time, the vehicle's real identity can be traced if necessary. 2) Navigation queries and results are protected to preserve user's confidentiality and operator's profit. On the other hand, one's real identity and navigation query are completely delinked using the idea of anonymous credential. 3) Information provided by RSUs can be properly authenticated in an efficient way.

We provide a security analysis and a simulation study to evaluate our scheme. Through the simulation, we find that a query can be completed (fulfilling all security requirements) in a reasonable amount of time. Also our scheme can lead to a savings of up to 55% in travelling time when compared with offline route searching approaches which do not take into account the real road conditions. Finally, through a partial implementation, we show that batch verification of signatures in RSUs is not desirable in our scheme as opposed to what was suggested in works like [7].

The rest of the paper is organized as follows: related work is reviewed in Section II. The system model and the problem statement are described in Section III. Some preliminaries are given in Section IV. Our schemes are presented in Section V. The analysis and evalution of our schemes are given in Sections VI, VII and VIII. The approach of batch verification is discussed in IX. Finally, Section X concludes the paper.

II. Related Work

The idea of real-time navigation using VANET is not totally new. A similar scheme is proposed in a recent work [8]. However, there are a number of differences between their scheme and ours. First, their scheme is of a small scale which covers a carpark while ours is large scale to cover the whole city and beyond. Second, in their scheme a carpark is monitored by three RSUs which take up the roles of determining a vehicle's location, searching for a vacant parking space and providing navigation service to guide the vehicle to go from the carpark entrance to the selected parking space. In our scheme, the road system in the city is monitored by a large number of RSUs which take up the navigation task in a distributed manner. Third, in terms of security functions, their scheme assumes RSUs to be fully trusted. This makes sense since the three RSUs are installed indoors and can be monitored by security guards. However, such an assumption is no longer valid in our outdoor setting. It is impossible to have security guards monitor all RSUs across the city. Thus, unlike their scheme, authentication of RSUs becomes a vital component in ours. Fourth, our scheme allows one's identity and navigation query to be delinked. This feature is only interesting for wide area navigation like ours. Thus, the scheme provided in [8] cannot be used to solve the navigation problem discussed in this paper. Besides, an application of real-time navigation is proposed in another recent work [6]. In addition to driving guidance, the returned routes are used for opportunistically routing multimedia information such as images and videos of a desired scene to vehicles.

Our scheme is based on the idea of indistinguishable (anonymous) credential. Such a credential system was introduced by Chaum [9]. The system allows a user to obtain a credential from one organization and later show the possession of the credential to another organization while the transactions at the two organizations are not linkable. The idea of anonymous credential has been adopted in different applications. For example, [10] proposes a credential-based privacy-preserving e-learning system under which a student can show his/her progress in e-learning without leaking his/her identity information.

In fact, VANET security is a hot research topic. Security issues and challenges of VANETs have recently been summarized by [11]. As early as 2007, a scheme called AMOEBA [12] was proposed to provide location privacy based on the concept of vehicle group navigation. In 2008, a number of works including [7], [13] and [14] were published to address different security issues in VANETs. In [7], a batch verification scheme known as IBV was proposed for an RSU to verify a large number of signatures at the same time using only three pairing operations. The scheme relies on a tamper-proof device to store an unchangeable master secret key. However, it can be expected that such a tamper-proof device will be compromised eventually (e.g. Infineon Trusted Platform Modules (TPMs) were compromised a few months ago [15]). And once one tamper-proof device is compromised, the whole system will be compromised. Thus in our VSPN scheme, we enable the master secret key to be updated regularly via RSUs, yet the RSUs still have no knowledge of it by means of the property of proxy re-encryption. In [13], an RSU-aided inter-vehicle communications scheme was proposed. A vehicle relies on an RSU to verify the signature of another vehicle. In [14], group communications in VANETs are considered and a group key update protocol was proposed.

In 2009, some security and privacy-enhancing communications schemes were proposed in [16]. Of particular interest, a group communications protocol was defined. After a simple handshaking with any RSU, a group of known vehicles can verify the signature of each other without any further support from RSUs. A common group secret is also developed for secure communications among group members. In the same year, a strategy was formulated for pseudonym update to sustain privacy when a vehicle is being observed by an adversary who has different capabilities [17]. Results show that by adopting the pseudonym update strategy, the privacy of a vehicle can be maximized. Recently in 2011, two more related works [18] and [19] were published. In [18], an efficient self-generated pseudonym mechanism based on Identity-Based Encryption (IBE) was proposed for protecting drivers' privacy. In [19], an efficient social-tier-assisted packet forwarding protocol STAP for achieving receiver-location privacy preservation in VANETs was proposed. The authors proposed to deploy storage-rich RSUs at social spots and let them form a virtual social tier. In this way, without knowing the receiver's exact location, a packet for him/her can first be forwarded and disseminated in the social tier concerned. Once the receiver visits one of social spots at a later time, he/she can receive the packet successfully.

Other recent efforts include [20] and [21]. These two works also target at driver privacy preservation but instead of using pseudo identities, the concept of group signature is adopted. The signature of any vehicle can be verified by the same group key but the actual signer can only be traced by a trusted party. Though privacy can be preserved, these schemes are rather complicated and may not be practical.

III. Problem Statement

A. System Model and Assumptions

Recall that a vehicular network consists of on-board units (OBUs) installed on vehicles, road-side units (RSUs) along the roads, and a trusted authority (TA). We assume the following:

1) TA is trusted but curious. It performs cryptographic operations such as key generation honestly but is curious about drivers' privacy such as navigation queries. To avoid being a single point of failure, redundant TAs which have identical functionalities and databases are installed.
2) TA and tamper-proof devices on vehicles are assumed to be trusted for the generation and management of anonymous credentials.
3) RSUs are not trusted and curious. Since they are placed along roadside, they can be easily compromised. Also they are curious about drivers' privacy such as navigation queries.
4) RSUs and TA communicate through a secure fixed network (e.g. Internet).
5) There exists a conventional public key infrastructure (PKI) for initial vehicle authentication. Each vehicle V_i having license plate number LP_i has a conventional public key CPK_i and a conventional private key CSK_i and is given a TA-signed certificate $Cert_i$ which contains CPK_i and LP_i. We will discuss details about the generation and verification of $Cert_i$ in Section V.
6) There also exists a conventional identity-based public key infrastructure (PKI) for TA and RSU authentication. The public key of the TA is the same as its real identity $TRID$ and is known by *everyone*. Also any RSU R_i broadcasts its public key which is the same as its real identity $RRID_i$ with hello messages periodically to vehicles that are travelling within its RSU-Vehicle Communications (RVC) range. Thus $RRID_i$ is known by all vehicles nearby. The validity of $RRID_i$ can be ensured using a certificate issued by the TA. We will discuss the details in Section V.
7) The real identity of any vehicle is only known by the TA and itself but not by others.
8) We assume that there is a reasonably large number of navigation queries issued to RSUs. Otherwise, if there is only one navigation query, the sender can be linked up with the query easily.
9) Each RSU has a local database storing road information in its range (e.g. GPS locations of boundaries, and names of buildings and streets) and how to get to its neighboring RSUs (e.g. distance and direction).
10) Each vehicle has a tamper-proof device which is responsible for all cryptographic-related functions such as storage of keys, generation of pseudo identities, signing messages and encryption of messages (details will be given one by one in the next section). Also its output interface is limited and we will specify that in the appropriate places in the next section. Finally, it is assumed

to have its own clock for generating correct time stamps and be able to run on its own battery [22]. Note that a vehicle can also have a conventional computer device for performing the verification of RSUs' hop information (to speed up the process and details will be given in the next section).

11) TA, RSUs and vehicle tamper-proof devices have roughly synchronized clocks. This can be done easily by requiring TA to periodically broadcast the current time to all vehicle tamper-proof devices via RSUs.

B. Adversary Model

We assume that an adversary is capable of performing the following without our VSPN scheme:

1) An adversary can issue or even impersonate any vehicle to issue navigation query into the system.
2) An adversary can trace the real identity of any vehicle and can reveal a vehicle's real identity by analysing multiple messages sent by it.
3) An adversary can obtain the content of any navigation query and navigation result by means of eavesdropping.
4) An adversary can link up a vehicle's query with its real identity by colluding with RSUs and TA.

Thus we aim at designing a scheme to prevent all these from happening.

C. Security Requirements

We aim at designing a scheme to provide VANET-based navigation to satisfy the following security requirements:

1) Message integrity and authentication: A vehicle should be authenticated before it can issue a navigation query. On the other hand, an RSU (vehicle) is able to verify that a message is indeed sent and signed by a certain vehicle (RSU) without being modified by anyone.
2) Identity privacy-preserving: The real identity of a vehicle should be kept anonymous from other vehicles as well as from RSUs and a third-party should not be able to reveal a vehicle's real identity by analysing multiple messages sent by it.
3) Traceability: Although a vehicle's real identity should be hidden from other vehicles and RSUs, TA should have the ability to obtain a vehicle's real identity so that the vehicle can be charged for using the navigation service. Also TA has the role to maintain liability via non-repudiation property of messages when accidents happen on the road.
4) Confidentiality: The content of a query and that of a navigation result should be kept confidential from eavesdroppers.
5) Unlinkability: Even if all RSUs and TA collude, they cannot link up a vehicle's query with its real identity.

Note that there can be other kinds of attacks such as distributed denial of service (DDoS) attacks in a VANET environment. However, there are already many existing techniques such as [23] and so we do not make it our focus in this paper.

IV. Preliminaries - Bilinear Maps

The section describes the concepts of bilinear maps and proxy re-encryption schemes in details.

A. Bilinear Maps

Our security schemes are *pairing-based* and defined on two cyclic groups with a mapping called *bilinear map* [24]. In this subsection, we briefly introduce what a bilinear map is.

Let $\mathbb{G}$ be a cyclic multiplicative group with generator g and $\mathbb{G}_{\mathbb{T}}$ be another. Both groups $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$ have the same prime order q. The mapping $\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ is called a bilinear map if it satisfies the following properties:

1) Bilinear: $\forall g_1, g_2, g_3 \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}$, $\hat{e}(g_2, g_1 \cdot g_3) = \hat{e}(g_1 \cdot g_3, g_2) = \hat{e}(g_1, g_2) \cdot \hat{e}(g_3, g_2)$. Also $\hat{e}(g_1^a, g_1^b) = \hat{e}(g_1, g_1^b)^a = \hat{e}(g_1^a, g_1)^b = \hat{e}(g_1, g_1)^{ab}$.
2) Non-degenerate: There exists $g_1, g_2 \in \mathbb{G}$ such that $\hat{e}(g_1, g_2) \neq 1_{\mathbb{G}_{\mathbb{T}}}$.
3) Computable: There exists an efficient algorithm to compute $\hat{e}(g_1, g_2)$ for any $g_1, g_2 \in \mathbb{G}$.

The bilinear map $\hat{e}$ can be constructed using pairings on elliptic curves. Each operation for computing $\hat{e}(g_1, g_2)$ is referred to as a pairing operation. Pairing operation is the most expensive operation in this kind of cryptographic schemes. The fewer the number of pairing operations, the more efficient the scheme is. The groups $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$ are called bilinear groups. The security of our schemes relies on the fact that the discrete logarithm problem (DLP) on bilinear groups is computationally hard, i.e., given the point $g_2 = g_1^a$, there exists no efficient algorithm to obtain a given g_1 and g_2. The implication is that we can transfer g_2 in an open wireless channel without worrying that a (usually some secret) can be known by any attackers.

B. Proxy Re-encryption Schemes

In our VSPN scheme, we make use of the properties of proxy re-encryption to let RSUs re-encrypt the most updated master secret s to vehicles while at the same time the RSUs do not know the value of s. In this subsection, we briefly introduce the concept of proxy re-encryption.

A proxy re-encryption scheme is similar to a traditional symmetric or asymmetric encryption scheme with the addition of a delegation function. The message sender can generate a re-encryption key based on his/her own secret key and the delegated user's key. A proxy can then use this re-encryption key to translate a ciphertext into a special form such that the delegated user can use his/her private key to decrypt the ciphertext. Two representative proxy re-encryption schemes can be found in [25] and [26].

The concept of proxy re-encryption is very useful in our VSPN scheme. In our scheme, we adopt an asymmetric approach. The TA first prepares a re-encryption key for each vehicle. RSUs can then use the re-encryption key to translate the encrypted master secret s into a form such that the vehicle concerned can decrypt using its private key. In this way, the master secret can be distributed by the RSUs while at the same time, it is kept secret from the RSUs.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

V. Our Solutions - VSPN

This section presents our VANET-based Secure and Privacy-preserving Navigation (VSPN) scheme. We first summarize our scheme into some basic steps (see Fig. 1):

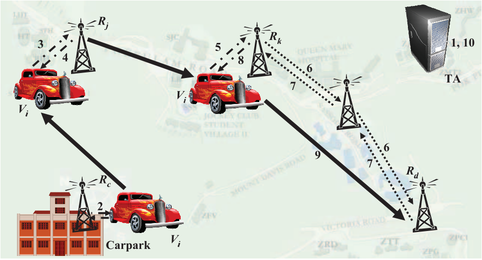

Fig. 1. Basic Steps in VSPN

1) TA sets up parameters and generates anonymous credentials.
2) Vehicle V_i's tamper-proof device starts up and requests for the master secret s from RSU R_c.
3) Vehicle V_i's tamper-proof device requests for a navigation credential from RSU R_j.
4) RSU R_j verifies V_i's identity and sends its tamper-proof device an anonymous credential.
5) After a random delay or after travelling for a random distance, V_i's tamper-proof device sends out its navigation request to RSU R_k.
6) RSU R_k forwards the navigation request to its neighbors. This process repeats until the request reaches RSU R_d covering the destination.
7) RSU R_d constructs the navigation reply message and sends it along the reverse path. Each hop along the path attaches the corresponding hop information (with signature).
8) RSU R_k forwards the navigation reply message to V_i's tamper-proof device which then verifies the messages from all RSUs along the route in a batch.
9) By presenting the navigation session number, each RSU along the route guides V_i to reach the next RSU closer to the destination.
10) Based on V_i's pseudo identity received from RSU R_j, TA reveals V_i's real identity for billing purpose.

Next we explain our scheme in details. The notations used in this paper are summarized in Table I.

TABLE I
Notations used in this paper

Symbol	Meaning
$\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$	Bilinear groups
g	Generator of $\mathbb{G}$
s	System master secret
$g_{pub} = g^s$	Public parameter
$TCPK$	Conventional public key of TA
$TCSK$	Conventional private key of TA
$TRID$	Identity of TA
TSK	Secret key of TA s.t. $TRID = g^{TSK}$
$TSIG_{TSK}(M)$	TA's signature on message M using TSK
R_i	RSU number i
RL_i	Location of RSU R_i
RC_i	Certificate of RSU R_i
$RRID_i$	Identity of RSU R_i
RSK_i	Secret key of RSU R_i s.t. $RRID_i = g^{RSK_i}$
C_T	Anonymous credential for period T
V_i	Vehicle number i
VC_i	Certificate of vehicle V_i
CPK_i	Conventional public key of vehicle V_i
CSK_i	Conventional private key of vehicle V_i
REK_i	Re-encryption key for vehicle V_i
$VRID_i$	Real identity of vehicle V_i
$VPWD_i$	Hardware activation password on V_i
$VPID_i$	Pseudo identity of vehicle V_i
VSK_i	Signing key of vehicle V_i
$S\_ENC_x(M)$	Symmetrical encryption of M using key x
$AS\_ENC_x(M)$	Asymmetrical encryption of M using key x
$SIG_x(M)$	Signature on message M using key x
$H(M)$	MapToPoint hash value [27] on message M
$h(M)$	One-way hash value of message M

A. Setup by TA

During system startup, the following will be carried out by TA.

- TA chooses $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$ that satisfy the bilinear map properties.
- Let g be the generator of $\mathbb{G}$. TA randomly picks $s \in \mathbb{Z}_q$ as the master secret and computes $g_{pub} = g^s$ as a public parameter. TA can update s and the corresponding g_{pub} at any time and the most updated s being encrypted using TA's public key (i.e. $AS\_ENC_{TRID}(s)$) is broadcasted to all RSUs while the most updated g_{pub} are made public. Such an update does not need to be carried out frequently. Instead, it is only needed when any vehicle unregisters (i.e. a vehicle is no longer eligible to know the value of s) or when any vehicle is proved to have been compromised (i.e. the value of s is already disclosed to attackers). Note that since s is encrypted using TA's public key, RSUs cannot know its value.
- TA assigns itself an identity $TRID$ and sets its secret key TSK such that $TRID = g^{TSK}$. $TRID$ is assumed to be known by everyone in the system.
- TA also generates a pair of conventional public and private keys, $TCPK$ and $TCSK$, only used for master key re-encryption purposes.
- For each RSU R_i located at RL_i, TA performs the following steps:
  - TA assigns it an identity $RRID_i$ and a secret key RSK_i such that $RRID_i = g^{RSK_i}$.
  - TA then generates R_i's certificate as $RC_i =< RRID_i, RL_i, TSIG_{TSK}(RRID_i||RL_i) >$ where $TSIG_{TSK}(RRID_i||RL_i)$ is TA's signature on the concatenation of $RRID_i$ and RL_i and is defined as $H(RRID_i||RL_i)^{TSK}$ where $H(.)$ is a MapToPoint hash function [27].
- During the first registration of each vehicle V_i, TA performs the following steps:
  - TA assigns each vehicle V_i a real identity $VRID_i$ and a tamper-proof device activation password

  $VPWD_i$. $VRID_i$ is defined as g^x where x is a random number. Note that TA does not need to keep the value of x after generating $VRID_i$.
  - TA also assigns each vehicle V_i a license plate number LP_i by coordinating with the transport department in the city.
  - TA generates a pair of conventional public and private keys, CPK_i and CSK_i respectively, for V_i.
  - TA generates a re-encryption key REK_i for V_i. REK_i is made from TA's conventional secret key $TCSK$ and V_i's conventional public and private keys, CPK_i and CSK_i, so that by having it, RSUs can translate a ciphertext encrypted using TA's public key $TCPK$ to a new ciphertext being decryptable by V_i.
  - TA signs V_i's certificate as $VC_i \; =< \; LP_i, CPK_i, REK_i, TSIG_{TSK}(LP_i||CPK_i||REK_i) \; >$ where $TSIG_{TSK}(LP_i||CPK_i||REK_i) = H(LP_i|| CPK_i||REK_i)^{TSK}$. Vehicle V_i can use VC_i for initial authentication to obtain the most updated master key s.
  - TA preloads $VRID_i$, $VPWD_i$, LP_i, CSK_i and VC_i into the tamper-proof device of V_i.

Throughout the paper, conventional asymmetric and symmetric encryptions and signatures are used occasionally. To make the context concise, let us use the notations $AS\_ENC_x(M)$ and $S\_ENC_x(M)$, $SIG_x(M)$ to denote asymmetrically encrypting, symmetrically encrypting and signing message M using the key x based on any asymmetric encryption, symmetric encryption , and signature algorithms, respectively.

B. Generation of Anonymous Credentials by TA

As mentioned in Section I, our scheme is based on the idea of anonymous credentials. Before we talk about how they are used, let us explain how they are generated by TA.

In our scheme, a navigation credential will expire after a predefined expiration period of time (e.g. a day). In other words, the navigation credentials on different days are different. Thus even if an attacker breaks the system and obtains a credential successfully, the impact to the system is limited.

Assume that we are now at time period T. TA performs two simple operations:

- TA computes the credential for the current period as $C_T \; =< \; \mathbf{NVC}, T, TSIG_{TSK}(\mathbf{NVC}||T) \; >$ where the keyword **NVC** is used to denote that it is a navigation credential and $TSIG_{TSK}(\mathbf{NVC}||T) = H(\mathbf{NVC}||T)^{TSK}$.
- TA sends $S\_ENC_s(C_T)$ to all RSUs securely via a fixed infrastructure.

From the definition of C_T, we can see that the credential carries no information about any user and that is why we call it "anonymous".

C. Activation and requesting for master key by vehicle tamper-proof device

When the vehicle V_i starts, the driver enters the real identity $VRID_i$ and password $VPWD_i$ (assigned by TA in Section V-A) into the tamper-proof device to activate it. Here only simple hardware checking is involved. Two cases are possible and the tamper-proof device reacts accordingly:

- If either the real identity or the password, or both are incorrect, the tamper-proof device refuses to perform further operations.
- If both the real identity and the password are correct, the tamper-proof device signs a master key request message as $SIG_{CSK_i}(\mathbf{MK\_Req})$. It then sends $< RRID_c, VC_i$, $SIG_{CSK_i}(\mathbf{MK\_Req}) >$ to an RSU R_c (with identity $RRID_c$) nearby.

Upon receiving from V_i, RSU R_c performs the following steps:

- It first verifies TA's signature on the certificate VC_i by checking whether the equality $\hat{e}(TSIG_{TSK} \quad (LP_i||CPK_i||REK_i), g) \quad = \hat{e}(H(LP_i||CPK_i||REK_i)^{TSK}, TRID)$ holds.
  Proof of correctness:
  $$\begin{aligned}&\hat{e}(TSIG_{TSK}(LP_i||CPK_i||REK_i), g)\\ &= \hat{e}(H(LP_i||CPK_i||REK_i)^{TSK}, g)\\ &= \hat{e}(H(LP_i||CPK_i||REK_i), g^{TSK})\\ &= \hat{e}(H(LP_i||CPK_i||REK_i), TRID) \qquad \square\end{aligned}$$
- If the TA's signature is valid, it proceeds to verify V_i's signature $SIG_{CSK_i}(\mathbf{MK\_Req})$ using the public key CPK_i as included in the certificate VC_i.
- If V_i's signature is valid also, RSU R_c proceeds to re-encrypt $AS_ENC_{TRID}(s)$ into $AS_ENC_{CPK_i}(s)$ using V_i's re-encryption key REK_i and sends $< CPK_i, AS\_ENC_{CPK_i}(s) >$ to V_i.

V_i's tamper-proof device can then use the private key CSK_i stored to decrypt and obtain s. Note that by default, V_i's temper-proof device does not provide any function for outputting the values of CSK_i or s. They are used for internal operations only.

On the other hand, our VSPN scheme supports vehicle revocation. The TA maintains a revocation list which contains certificates of all revoked vehicles (e.g. those vehicles which have been proved to have committed any kind of attacks to the system). This revocation list is then broadcasted to all RSUs. Having this mechanism, RSUs will not send the encrypted master secret to revoked vehicles in order to protect the system.

D. Requesting for anonymous credential by vehicle tamper-proof device

In order to obtain anonymous credentials, V_i's tamper-proof device performs the following steps:

- It first generates a pseudo identity $VPID_i$ which is composed of two parts $VPID_{i1}$ and $VPID_{i2}$ (or denoted as $(VPID_{i1}, VPID_{i2})$). These two parts are defined as $VPID_{i1} = g^r$ and $VPID_{i2} = VRID_i \oplus H(g^r_{pub})$, where r is a per-session random nonce, respectively.

- It then composes the navigation credential request message $M_i = \{\textbf{NVC\_REQ}\}$.
- It also picks a random number $rand$ and encrypts it using R_j's identity as $AS\_ENC_{RRID_j}(rand)$. This random number becomes a shared secret between itself and RSU R_j. R_j will use it to encrypt the credential at a later stage.
- Next it generates the signing key VSK_i as $(VSK_{i1}, VSK_{i2}) = (VPID_{i1}^s, HP_i^s)$ where $HP_i = H(VPID_{i1}||VPID_{i2})$.
- It generates its signature σ_i on M_i and T_i (T_i is the current timestamp given by the tamper-proof device) as $VSK_{i1} \times VSK_{i2}^{h(M_i||T_i)}$ where $h(.)$ is a one-way hash function such as SHA-1 [28].
- Finally it sends $< RRID_j, AS\_ENC_{RRID_j}(rand)$, $VPID_i$, $M_i, T_i, \sigma_i >$ to RSU R_j nearby.

RSU R_j then performs the following steps:

- It first checks the timestamps in the messages. For any message, if the difference between the attached timestamp and the current time is larger than a threshold (which is a system parameter), the message is ignored. This can help reduce the impact of replay attack.
- It then verifies V_i's signature by checking whether the equality $\hat{e}(\sigma_i, g) = \hat{e}(VPID_{i1} \times HP_i^{h(M_i)}, g_{pub})$ holds.
  Proof of correctness:
  $\hat{e}(\sigma_i, g)$
  $= \hat{e}(VSK_{i1} \times VSK_{i2}^{h(M_i)}, g)$
  $= \hat{e}(VPID_{i1}^s \times HP_i^{sh(M_i)}, g)$
  $= \hat{e}(VPID_{i1} \times HP_i^{h(M_i)}, g^s)$
  $= \hat{e}(VPID_{i1} \times HP_i^{h(M_i)}, g_{pub})$ □
- If V_i's signature is valid, R_j encrypts the encrypted anonymous credential for the current period $S\_ENC_s(C_T)$ using $rand$ and sends $< VPID_i, S\_ENC_{rand}(S\_ENC_s(C_T)) >$ back to it. Note again that $rand$ provides a secure communications channel between R_j and V_i.

Upon receiving $S\_ENC_{rand}(S\_ENC_s(C_T))$, V_i's tamper-proof device decrypts it using $rand$ and s in order and stores the anonymous credential C_T. Note that by default, V_i's temper-proof device does not provide any function for outputting the anonymous credential C_T and it is used for composing messages only. This helps to prevent a vehicle from illegally transferring an anonymous credential to another unauthorized party.

E. Requesting for navigation service by vehicle tamper-proof device

Next let us come to the core part of our scheme - requesting for navigation service. Note that if V_i obtains the credential C_T from RSU R_j and if it sends out its navigation query to R_j immediately, its real identity and its query may be linked up if R_j colludes with TA (since TA can always recover V_i's real identity from its pseudo identity based on our traceability requirement), especially when V_i is the only vehicle which requests credential from R_j. We propose three simple approaches to avoid this from happening:

1) V_i pre-requests a number of navigation credentials before they are being used. For example, if a driver knows that he/she will require navigation service some time in the day, he/she can pre-request an appropriate number of navigation credentials early in the morning before the vehicle starts any journey.
2) V_i sends out its navigation query to R_j only after a random delay. This is because under normal situation, there will be credential requests from other vehicles during that random period and as a result R_j cannot link up which query belongs to which credential request.
3) V_i sends out its navigation query at another RSU (say $R_k \neq R_j$) after travelling for a random distance. Since R_k does not know V_i's credential request (thus pseudo identity), even if it colludes with TA, it cannot link up V_i's real identity and its query.

Now without loss of generality, assume that V_i sends its navigation request to RSU R_k. V_i's tamper-proof device performs the following steps:

- It first composes the navigation request message $M_i = \{\textbf{NV\_REQ}, DEST\}$ where $DEST$ can be anything representing the destination desired (e.g. GPS coordinates).
- It picks a random number $rand$ which is for R_k to encrypt the navigation result at a later stage.
- It then requests TA for a navigation session number nsn. To avoid the collision of navigation session numbers in different navigation instances, navigation session numbers are centrally generated by TA. TA maintains a list containing all used navigation session numbers. Whenever TA is requested by any tamper-proof device for a new navigation session number, TA randomly picks one which is not on the list. TA also periodically flushes the list by removing the earliest entries.
- It also retrieves C_T from its local storage and sends $< RRID_k, AS\_ENC_{RRID_k}(rand, nsn, C_T, M_i) >$ to R_k.
- It then stores $rand$ and nsn locally.
- After that V_i's tamper-proof device deletes C_T from its storage so that a credential will not be used more than once. In case the driver wants to make another navigation request on the same day, he/she has to request for another credential via its tamper-proof device and this in turn leads to another service charge.
- V_i keeps the session alive with R_k until it receives the navigation reply.

Upon receiving V_i's request message, RSU R_k performs the following:

- It decrypts the message using its private key.
- It ensures that the credential used C_T is not outdated (e.g. the timestamp should be within a pre-defined number of expiration periods before the current time period).
- It then verifies TA's signature on C_T (having the for-

mat $< \mathbf{NVC}||T, TSIG_{TSK}(\mathbf{NVC}||T)>$) by checking whether $\hat{e}(TSIG_{TSK}(\mathbf{NVC}||T), g)$ is equal to $\hat{e}(H(\mathbf{NVC}||T), TRID)$.
Proof of correctness:
$\hat{e}(TSIG_{TSK}(\mathbf{NVC}||T), g)$
$= \hat{e}(H(\mathbf{NVC}||T)^{TSK}, g)$
$= \hat{e}(H(\mathbf{NVC}||T), g^{TSK})$
$= \hat{e}(H(\mathbf{NVC}||T), TRID)$ □

- If the signature is valid, it proceeds to the route searching process.
- It stores $rand$ and nsn locally for later use.

F. Navigation request and reply propagation among RSUs

In this sub-section, let us look at how V_i's navigation query is propagated across the network of RSUs and how the result is sent back to V_i. RSU R_k takes up the role of initiating the route searching process by composing the route request message $M_k = \{\mathbf{RT\_REQ}, nsn, RRID_k, DEST\}$ and broadcasts it to all neighbors which are closer to $DEST$ than itself.

Any receiving RSU performs the following steps:

- It first stores nsn (the navigation session number), $RRID_k$ and $DEST$ into its navigation routing table to build up the reverse path so that it can send any reply back to R_k later on.
- It then checks whether $DEST$ is within its range.
- If $DEST$ is not within its range, it simply re-broadcasts M_k to all neighbors which are closer to $DEST$ than itself.
- If $DEST$ is within its range, it computes the route reply message $M_d = \{\mathbf{RT\_RPY}, nsn, RRID_d, RL_d, RC_d, HopInfo_d\}$ and sends it back to its previous RSU hop. Here $HopInfo_d =< AvgSpd_d, RoadCond_d, \sigma_d >$. $AvgSpd_d$ represents the average vehicle speed in its range. Note that to better reflect the flow status of a road, $AvgSpd_d$ is taken as an average value over a pre-defined period (say 30 minutes). $RoadCond_d$ represents the summarized road conditions in its range. The size of $RoadCond_d$ can be very small since we can use some pre-agreed symbols to represent common road conditions such as traffic jam, collision, fire, etc. σ_d is R_d's signature on the concatenation of $AvgSpd_d$ and $RoadCond_d$ and is defined as $H(AvgSpd_d||RoadCond_d)^{RSK_d}$.

Each RSU hop along the reverse path R_{im} repeats the steps done by R_d and includes information corresponding to its hop (i.e. $RRID_{im}, RL_{im}, RC_{im}, HopInfo_{im}$) into the route reply message. R_{im} also stores the next hop of the forward path (i.e. the identity of the RSU from which it receives the route reply message) into its routing table for guiding V_i later on.

Now let us go back to R_k, the RSU which initiates the route searching process. Upon receiving a navigation reply, R_k will not forward it to vehicle V_i immediately. Instead, it waits for a threshold (which is a system parameter) amount of time for more replies (possibly from RSUs on other directions). R_k then performs the following:

- It verifies each RSU hop's information (the verification procedure is the same as the one used by vehicle V_i and we will discuss the details in the next sub-section).
- It picks the travelling route that has the highest average speed and does not contain any unusable road (e.g. those totally blocked by a traffic accident).
- It then encrypts all RSU hops' information using $rand$ and forwards it to vehicle V_i. Note that R_k includes nsn into its message for V_i so that other vehicles nearby know that they do not need to process the message.

G. Verification of RSUs' hop information by vehicle tamper-proof device

Recall that the reply contains a set of identities, a set of locations, a set of certificates and a set of hop information (average speed and road condition together with signatures), each corresponding to an RSU along the route returned. To verify the average speed and road condition provided by an RSU, its signature is verified using its identity. In turn, to verify an RSU's real identity, its certificate has to be verified using TA's identity. Note that the verification process may take excessive amount of time if carried out by a tamper-proof device with today's technology. As such, this part can be relaxed to be carried out by a conventional car computer device in order to speed up the process.

V_i's tamper-proof device follows the following steps to verify RSUs' hop information:

- It first decrypts R_k's reply using the stored $rand$.
- It then verifies RSUs' certificates. Without loss of generality, assume the RSUs along the returned route have real identities $RRID_{first}$, ..., $RRID_{last}$, locations RL_{first}, ..., RL_{last} and TA signatures $TSIG_{TSK}$ $(RRID_{first}||RL_{first})$, ..., $TSIG_{TSK}(RRID_{last}|| RL_{last})$. Vehicle V_i can then verify each of the $(last - first + 1)$ signatures (say signature $TSIG_{TSK}(RRID_i||RL_i)$ for the RSU located at RL_i with real identity $RRID_i$) by checking whether $\hat{e}(TSIG_{TSK}$ $(RRID_i||RL_i), g) = \hat{e}(H(RRID_i||RL_i), TRID)$
Proof of correctness:
$\hat{e}(TSIG_{TSK}(RRID_i||RL_i), g)$
$= \hat{e}(H(RRID_i||RL_i)^{TSK}, g)$
$= \hat{e}(H(RRID_i||RL_i), g^{TSK})$
$= \hat{e}(H(RRID_i||RL_i), TRID)$ □
- Next it verifies the signature by each of these $(last - first + 1)$ RSUs. Assume that these $(last - first + 1)$ RSUs provide the average speeds $AvgSpd_{first}$, ..., $AvgSpd_{last}$, road conditions $RoadCond_{first}$, ..., $RoadCond_{last}$ together with signatures σ_{first}, ..., σ_{last}. Vehicle V_i verifies each of these signatures (say signature σ_i corresponding to average speed $AvgSpd_i$ and road condition $RoadCond_i$) by checking whether $\hat{e}(\sigma_i, g) = \hat{e}(H(AvgSpd_i||RoadCond_i), RRID_i)$.
Proof of correctness:
$\hat{e}(\sigma_i, g)$
$= \hat{e}(H(AvgSpd_i||RoadCond_i)^{RSK_i}, g)$

$$= \hat{e}(H(AvgSpd_i||RoadCond_i), g^{RSK_i})$$
$$= \hat{e}(H(AvgSpd_i||RoadCond_i), RRID_i) \qquad \square$$

Note that for both verifications, vehicle V_i can actually perform batch verification to save the number of pairing operations. We will present the details in Section IX.

H. Guiding to destination by RSUs

Having the returned route, if V_i has GPS device installed and it can receive GPS signals for current location, it can simply search for each RSU based on the list of RL_i. However, GPS device is not an assumption of our scheme. Even if V_i does not have GPS device installed, our scheme can make use of the VANET to guide V_i to the destination.

To use the guiding service, V_i first generates a random number $rand$ and sends $< RRID_k, AS\_ENC_{RRID_k}(rand, nsn) >$ to R_k. Here nsn is the navigation session number generated earlier and R_k is the first RSU along the route.

Upon receiving the message from V_i, R_k performs three simple steps:

- It decrypts the message using its private key to obtain $rand$ and nsn.
- It then searches its navigation routing table to dig out the session with session number nsn and sends V_i information (e.g. direction) about how to get to the next RSU hop along the forward path (or the destination if it is already the last hop).
- To avoid being eavesdropped by vehicles nearby, the information is symmetrically encrypted by $rand$.

V_i repeats this process for each RSU hop along the route until it reaches the destination desired.

I. Urgent change of route initiated by RSU

Road conditions vary abruptly. A road which is initially in good condition may become blocked in a second. Thus our scheme is designed in such a way that the querying vehicle V_i will be informed about important changes in road conditions along the returned route.

Assume RSU R_b is an RSU along the returned route. Now if a road within its range is blocked, it immediately composes the road blocking notification message which is defined as $M_b = \{\textbf{ROAD\_BLOCKED}\}$ and sends $\{M_b, nsn, RRID_b, RL_b, RC_b, RSIG_{RRID_b}(M_b)\}$ to the next RSU hop along the reverse path. The message is propagated along the reverse path until an RSU that is currently in contact with V_i is reached. That RSU forwards the message to V_i. Again that RSU includes nsn into its message for V_i so that other vehicles nearby know that they do not need to process the message. V_i's tamper-proof device then verifies R_b's certificate and signature using the methods in Section V-G. After that, V_i can initiate a new navigation query to seek for an alternative route.

J. Traceability of real identity by TA

With V_i's pseudo identity $VPID_i = (VPID_{i1}, VPID_{i2}) = (g^r, VRID_i \oplus h(g^r_{pub}))$ and the master secret s, TA can retrieve V_i's real identity by computing $VRID_i = VPID_{i2} \oplus h(VPID^s_{i1})$.

VI. Security Analysis

In this section, we briefly analyze our scheme with respect to the security requirements listed in Section III.

1) Message integrity and authentication: For both TA's and RSUs' signatures, we adopt the Boneh-Lynn-Shacham (BLS) signature scheme and its security has been proved formally in [29]. Basically, TA's signature on message M is defined as $H(M)^{TSK}$. Since TSK is only known by TA, no others can forge the signature. Similarly, RSU R_j's signature on message M is defined as $H(M)^{RSK_j}$. Again since RSK_j is only known by R_j, no others can forge the signature.
   Vehicle V_i's signature is composed of VSK_{i1} and VSK_{i2}. VSK_{i1} is defined as g^{rs}. We argue that if the asymmetric encryption scheme $AS\_ENC_x(M)$ adopted by us is secure and if Diffie-Hellman (DH) problem is hard, then a vehicle's message cannot be forged by an attacker and our scheme is secure against existential forgery, adaptive chosen message attack under random oracle model. The proof is as follows.
   An RSU transmits the value of s to a tamper-proof device in encrypted form $AS\_ENC_{CPK_i}(s)$ where CPK_i is the conventional public key of the vehicle concerned and the corresponding conventional private key CSK_i is securely stored in its tamper-proof device. This tamper-proof device does not provide any function for outputting the values of CSK_i or s and they are used for internal operations only. Thus if the asymmetric encryption scheme $AS\_ENC_x(M)$ we adopted is secure, an attacker has no way to break the ciphertext $AS\_ENC_{CPK_i}(s)$ and obtain s. Note that the proxy re-encryption scheme we adopted ensures that RSUs do not know the value of s. Thus even an attacker collude with any RSU, it cannot gain any advantage.
   Next we show that if DH is hard, then a vehicle's message cannot be forged by an attacker and our scheme is secure against existential forgery, adaptive chosen message attack under random oracle model. The proof can be found in Appendix I.
2) Identity privacy preserving: In this sub-section, we show that an attacker cannot obtain a vehicle's real identity easily. Since the only information that is related to a vehicle's real identity and is exposed in the network is its pseudo identity, we show that an attacker cannot obtain a vehicle's real identity even it is keeps its pseudo identity. We argue that if Decision Diffie Hellman (DDH) problem is hard, then the pseudo identity of a vehicle can preserve its real identity. The proof can be found in Appendix II.
   On the other hand, the random nonce r makes the pseudo identity of a vehicle different in different messages. This makes tracing the location of a particular vehicle over time even more difficult.

To conclude, to trace the real identity, one needs to know the value of s but s is only known by all tamper-proof devices and TA. A tamper-proof device (which can prevent unauthorized parties from modifying its logic or reading its stored data) is not supposed to carry out such a traceability function. Thus no one except TA can trace the real identity of a particular vehicle and privacy is preserved.

3) Traceability: Section V-J shows that TA is able to trace a vehicle's real identity, thus traceability is satisfied.
4) Confidentiality: First we consider the anonymous credential. When vehicle V_i requests for a navigation credential from RSU R_j, it first picks a random number $rand$ and securely sends it to R_j. R_j in return encrypts the encrypted credential using $rand$. Thus if the symmetrical encryption scheme $S\_ENC_x(M)$ we adopted is secure, neighboring vehicles cannot illegally receive the encrypted credential $S\_ENC_s(C_T)$ by eavesdropping messages from the air.
Similarly, when vehicle V_i requests for navigation service from RSU R_k, though the credential C_T is going out of the tamper-proof device, it is included in the encrypted block $< AS\_ENC_{RRID_k}(rand, nsn, C_T, M_i) >$. Thus if the asymmetrical encryption scheme $AS\_ENC_x(M)$ we adopted is secure, neighboring vehicles cannot illegally receive the credential C_T by eavesdropping messages from the air.
Next we consider the navigation query. V_i encrypts its query using RSU's identity (as included in the encrypted block $< AS\_ENC_{RRID_k}(rand, nsn, C_T, M_i) >$). If the asymmetrical encryption scheme $AS\_ENC_x(M)$ we adopted is secure, it is kept confidential from others.
Finally, we consider the navigation result. When vehicle V_i requests for navigation service from RSU R_k, it picks another random number and R_k in return encrypts the navigation result using that random number. Again if the symmetrical encryption scheme $S\_ENC_x(M)$ we adopted is secure, no other vehicles can eavesdrop the route even if they want to go to the same destination. The profit of the operator is thus protected.
5) Unlinkability: As discussed in SectionV-E, we have three approaches to avoid TA and RSUs linking up a vehicle's identity and navigation query sent by it. A driver can pre-request a number of navigation credentials before they are being used. On the other hand, after a driver obtains an anonymous credential, it can present it to the same RSU after a random time interval or to a different RSU for navigation service. Note that the anonymous credentials given to all vehicles are identical within a period. Let us consider these approaches one by one.

   a) In the first approach, V_i pre-requests a number of navigation credentials before they are being used. Assume that there are credential requests from N vehicles, the probability of linking a vehicle's pseudo identity and navigation query sent by it is only $1/N$.
   b) In the second approach, V_i presents its navigation query to the same RSU which it requests for credential after a random delay. Assume that there are credential requests from $(N-1)$ other vehicles during this random period, the probability of linking a vehicle's pseudo identity and navigation query sent by it is only $1/N$.
   c) In the third approach, V_i presents its navigation query to another RSU. Since this later RSU does not know V_i's pseudo identity and identity verification is based on the anonymous credential, it can link up V_i's query with its identity only if it colludes with TA. In that case, assume that there are credential requests from N vehicles, the probability of linking a vehicle's pseudo identity and navigation query sent by it is only $1/N$.

Thus linkability can be minimized especially when more vehicles use the navigation service.

VII. Analysis on Time Complexity

In this section, we briefly analyze the time complexity of our VSPN scheme. Note that we ignore the time complexity involved in setup since it can be done offline and is only done once occasionally (e.g. when TA wants to update the public parameters). It is not critical to the efficiency of our VSPN scheme.

We let T_{mul} denote the time required to perform one point multiplication over an elliptic curve, T_{mtp} denote the time required to perform one MapToPoint hash function [27], and T_{par} denote the time required to perform one pairing operation. We further let T_{aenc}, T_{adec}, T_{senc}, T_{sdec}, T_{csig} and T_{renc} denote the time required to perform asymmetric encryption, asymmetric decryption, symmetric encryption, symmetric decryption, conventional signature and reencryption operations respectively. As argued by [7], these operations dominate the speed of signature generation and signature verification, we only consider the time taken by these operations and neglect all others such as addition, scalar value manipulation and one-way hash function. We consider the experiment in [30] for an MNT curve [31] with embedding degree $k = 6$, G being represented by 161 bits and order q being represented by 160 bits, on an Intel Pentium IV 3.0 GHz machine. The following results are obtained: T_{mul} = T_{mtp} = 0.6 ms, T_{par} = 4.5 ms.

Let us consider the steps in our VSPN scheme one by one. According to Section V-B, TA takes $T_{mul} + T_{mtp} + T_{senc}$ of time to generate the navigation credential for the current period. According to Section V-C, vehicle V_i takes T_{csig} of time to sign the master key request message. Next the RSU nearby takes $T_{mul} + T_{mtp}$ of time to verify V_i's certificate and then takes T_{renc} of time to re-encrypt the master key for V_i. According to Section V-D, when vehicle V_i requests for an anonymous credential, it takes $5T_{mul} + 2T_{mtp}$ of time for generating a signature (2 T_{mul} and 1 T_{mtp} for computing pseudo identity, 2 T_{mul} and 1 T_{mtp} for computing signing key and 1 T_{mul} for computing signature). The RSU nearby then takes $2T_{par} + T_{mtp} + T_{mul}$ or $2T_{par} + nT_{mtp} + nT_{mul}$ of time for single or batch (n) signature verification. Finally, the RSU takes $2T_{senc}$ of time to encrypt the credential for the current session for V_i. According to Section V-E, when vehicle V_i requests for navigation service, it takes T_{aenc} of time to encrypt the request message. The RSU nearby then takes T_{adec} of time to decrypt the message and then takes T_{mtp} of time to verify the anonymous credential presented by V_i. According to Section V-F, each RSU hop takes $T_{mul} + T_{mtp}$ of time to sign its hop information. According to Section V-G, vehicle V_i takes $2T_{mtp}$ of time to verify each RSU hop's certificate and signature on hop information. According to Section V-H, vehicle V_i takes T_{aenc} of time to generate the guiding service request message to an RSU nearby. According to Section V-I, if a road within an RSU's range is blocked, that RSU takes $T_{mul} + T_{mtp}$ of time to sign a road blocking message. According to Section V-J, TA can trace a vehicle's real identity in T_{mul} of time.

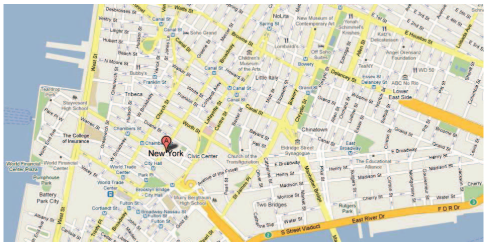

Fig. 2. City Road System in New York

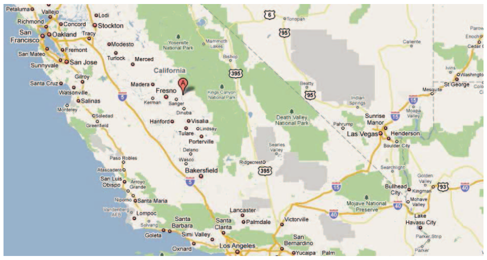

Fig. 3. Countryside Highway System in California

VIII. Simulation Results

In this section, we evaluate our VSPN scheme in terms of processing delay (including those imposed by cryptographic operations in our scheme) and the reduction in travelling time using a network simulation program. Through simulation, we show that the processing delay caused by our cryptographic functions is minimal while the savings in travelling time after using our scheme is significant. Note that since the generation of anonymous credentials can be done separately offline, we do not consider it in our simulation.

A. Simulation Models

In our simulation, we made use of two maps downloaded from the TIGER database [32] - one is New York and the other is California. New York represents a city road system (see Fig. 2 for the Google Map [33]) in which most roads have speed limit of 50 km/h. California, on the other hand, represents a countryside road system (see Fig. 3 for the Google Map [33]) in which some highways have speed limit up to 120 km/h. For New York, we took 14498 roads into consideration and placed 8477 RSUs onto them. For California, we took 11668 roads into consideration and placed 8948 RSUs onto them. The RSUs are placed in such a way that there is at least one RSU covering the two ends of each road since V2I communication is more critical there. Other RSUs are then randomly placed to improve the coverage. With the consideration of speeding behavior, we assume New York has average vehicle speed readings from 0 km/h (road blocking situation) to 70 km/h (speeding situation) while California has average vehicle speed readings from 0 km/h (highway blocking situation) to 140 km/h (speeding situation).

Some of the settings and parameters of our simulation are adopted from [13], [7] and [16]. The RSU-to-Vehicle Communication (RVC) and the Inter-Vehicle Communication (IVC) ranges are set to 600 m and 300 m, respectively. In the backbone, there is a TA server. RSUs communicate with each other and with TA via a fixed infrastructure. The bandwidth of the DSRC channel and the fixed infrastructure are assumed to be 6 Mb/s and 10 Mb/s, respectively. Regarding processing time, following the experiment on an Intel Pentium IV 3.0 GHz computer in [30], we assume that each pairing operation takes 4.5 ms and each point multiplication over an elliptic curve takes 0.6 ms. Each conventional asymmetric encryption takes 1.2 ms while each conventional symmetric encryption takes only 0.6 ms. In our VSPN scheme, an RSU needs to look up its routing table for forwarding direction. We assume such look-up can be accomplished in 0.6 ms on average.

Following [13] and [7], the size of some message components are fixed in our simulation: 42 bytes for pseudo identity, 21 bytes for ECC-type signature and 21 bytes for ECC-type public key. We further fix the size of those components that are newly introduced in our schemes: 5 bytes for control messages like **NVC_REQ**, 20 bytes for each representation of GPS location, 255 bytes for timestamp and 10 bytes for random number.

For each map, we define a fixed number of geographical distance ranges. For each range, we randomly pick 60 sets of sources and destinations that are within the geographical distance range. We treat them as the current location and the desired destination of a navigation querying vehicle respectively. When the experiment starts, about 10% of all roads are blocked. We only consider sources and destinations that have roads connected and these roads are not blocked at this time. Without loss of generality, we assume that a vehicle requests for a navigation credential or sends out its navigation query once it enters an RSU's range (upon hearing its beacon broadcasts). Since a vehicle can wait for a random delay or travel for a random distance after obtaining a navigation credential before sending out its navigation query, we define the processing time as the period from when the vehicle sends out its navigation query to when it finishes verifying the information provided by all RSUs along the returned path. This processing time is then normalized by the duration that the vehicle is in the range of the RSU to which it sends its query. Here we assume the vehicle concerned keeps on moving without being blocked by traffic jam or accident.

Besides processing time, we also introduce another terminology known as worst case urgent notification time. Upon a vehicle finding a route to a desired destination, we assume that a road blocking condition suddenly appears and as a result, the returned route can no longer be used. At that moment, the RSU covering the destination immediately sends an urgent notification message to the vehicle so that another route request can be made. Hence we define the worst case urgent notification time as the period from when the RSU covering the destination sends out the urgent notification message to when the vehicle finishes verifying the message. We claim that this is the worst case because the distance between the vehicle and the RSU concerned is the furthest at that time. To compare it with the processing time, we normalize this urgent notification time in the same way as before.

Next, we compare the travelling time of the route returned by our scheme and that by the offline map data searching approach (with and without the help of TMC service). The route returned by the offline map data searching approach is basically the shortest distance route. For the approach with TMC service, a driver can make use of TMC broadcast information to learn whether a returned route is blocked or not.

For all the above, the data from 15 sets are averaged to obtain a data point as shown in Fig. 4 to 7 below. Note that in all the figures, we abstract a range along the x-axis by its middle point (i.e. its class mark).

Besides processing delay, worst case urgent notification time and travelling time, we also evaluate the route blocking rate. For each range, we randomly pick 100 sets of sources and destinations that are within the geographical distance range. Different from before, we consider sources and destinations that have roads connected but these roads can be blocked when the experiment starts. Among the 100 cases, we evaluate the probability that the querying vehicle cannot reach the desired destination (i.e. the route found is indeed blocked and can cause a long travelling delay) by using our VSPN scheme and the offline map data searching approach (with and without the help of TMC service).

B. Simulation Results

In the first set of experiments, we consider the map of New York. We consider 20 geographical distance ranges of 1 km each. That is, the closest source and destination we pick are only 1 km apart while the furthest are 20 km.

From Fig. 4, we can see that the processing time increases with geographical distance. When the source and the destination nodes are further away, more RSU hops are involved. This not only leads to more RSU signing operations but also more pairing operations at the vehicles in the verification phase. Nevertheless, among all geographical distance ranges, the processing time is less than 1.4 % of the duration that the vehicle stays in the querying RSU's range. Thus there is sufficient time for the vehicle to finish its navigation query and to verify the result. The same figure also shows the worst case urgent notification time. Among all geographical distance ranges, the worst case urgent notification time is less than 0.1 % of the duration that the vehicle stays in the querying RSU's range. Thus even if a returned route is found blocked, there is sufficient time for the vehicle to request for an alternative route.

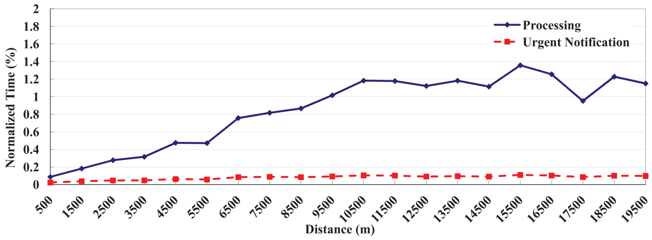

Fig. 4. Normalized Processing Time and Worst Case Urgent Notification Time vs. Geographical Distance (New York)

Fig. 5 shows the travelling time comparison between our VSPN scheme and offline map data searching approaches. As the geographical distance increases, the travelling time increases under both schemes. For all geographical distance ranges, the travelling route returned by our VSPN scheme gives lower delay than that returned by the offline map data checking approach (even with the help of TMC service). The gap increases as the displacement increases. The difference can be up to 13 minutes (a gain of 39 %).

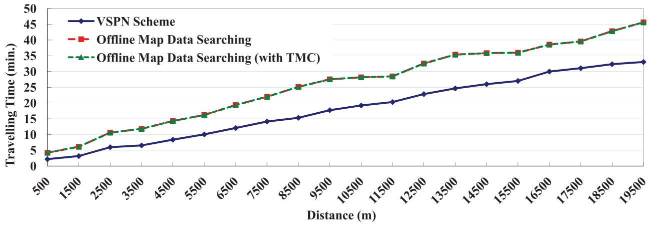

Fig. 5. Travelling Time vs. Geographical Distance (New York)

The route blocking rate is shown in Fig. 6. Interestingly for all 20 geographical distance ranges, the offline map data searching approach without the help of TMC service gives much higher blocking rate than our VSPN scheme and the approach with the help of TMC service. The reason is that offline map data searching does not consider real time road conditions at all and the returned route is only geographically shortest. With the help of TMC service, the blocking rate becomes comparable to ours.

In the second set of experiments, we consider the map of California. We consider 16 geographical distance ranges of 5

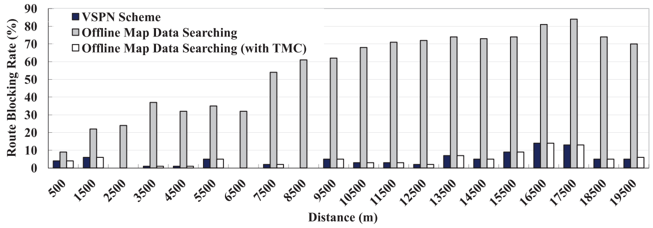

Fig. 6. Route Blocking Rate vs. Geographical Distance (New York)

km each. That is, the closest source and destination we pick are 5 km apart while the furthest are 80 km.

From Fig. 7, we can see that as the geographical distance increases, the processing time increases. The reason is the same as in the New York case. Nevertheless, even for the furthest source and destination points, the processing time is only 3.3 % of the duration that the vehicle stays in the querying RSU's range. Thus there is sufficient time for the vehicle to finish the navigation query and to verify the returning result. The same figure also shows the worst case urgent notification time. Among all geographical distance ranges, the worst case urgent notification time is only less than 0.3 % of the duration that the vehicle stays in the querying RSU's range. Thus even a returned route is found blocked finally, there is sufficient time for the vehicle to request for an alternative route. Since roads in California are longer than those in New York, the processing time and urgent notification time for the California case are double that for the New York case.

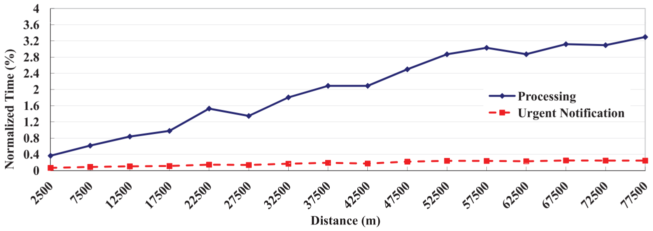

Fig. 7. Normalized Processing Time and Worst Case Urgent Notification Time vs. Geographical Distance (California)

From Fig. 8, we can see that the travelling time increases with geographical distance. For all geographical distance ranges, the travelling route returned by our VSPN scheme gives lower delay than that returned by the offline map database checking approach (even with the help of TMC service). The gap can be up to 33 minutes (a gain of 55 %). Again since roads in California are longer than those in New York, the travelling time for the California case is double that for the New York case.

The route blocking rate is shown in Fig. 9. Interestingly for all 16 geographical distance ranges, the offline map data searching approach without the help of TMC service gives much higher blocking rate than our VSPN scheme and the approach with the help of TMC service. As in the New York case, with the help of TMC service, the blocking rate becomes comparable to ours.

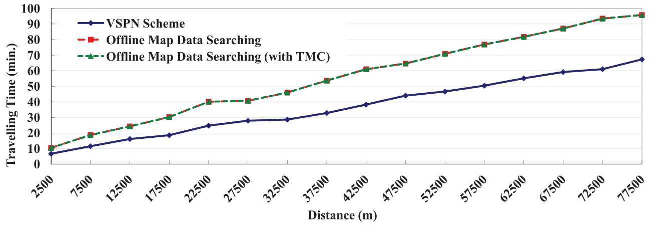

Fig. 8. Travelling Time vs. Geographical Distance (California)

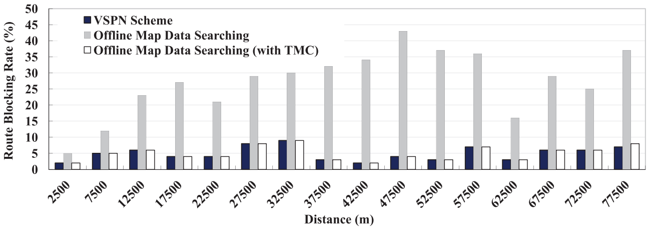

Fig. 9. Route Blocking Rate vs. Geographical Distance (California)

IX. Batch Verification Approach

In this section, we discuss how batch verification can be done when a vehicle needs to verify a set of RSUs in Section V-G.

Let us reconsider how the RSUs' certificates can be verified. Without loss of generality, assume the RSUs along the returned route have real identities $RRID_{first}$, ..., $RRID_{last}$, locations RL_{first}, ..., RL_{last} and TA signatures $TSIG_{TSK}(RRID_{first}||RL_{first})$, ..., $TSIG_{TSK}(RRID_{last}||RL_{last})$. Vehicle V_i can then verify the $(last - first + 1)$ signatures in a batch by checking whether $\hat{e}(\prod_{i=first}^{last} TSIG_{TSK}(RRID_i||RL_i), g)$
$= \hat{e}(\prod_{i=first}^{last} H(RRID_i||RL_i), TRID)$

Proof of correctness:

$\hat{e}(\prod_{i=first}^{last} TSIG_{TSK}(RRID_i||RL_i), g)$
$= \hat{e}(\prod_{i=first}^{last} H(RRID_i||RL_i)^{TSK}, g)$
$= \hat{e}((\prod_{i=first}^{last} H(RRID_i||RL_i))^{TSK}, g)$
$= \hat{e}(\prod_{i=first}^{last} H(RRID_i||RL_i), g^{TSK})$
$= \hat{e}(\prod_{i=first}^{last} H(RRID_i||RL_i), TRID)$ □

Further assume these $(last - first + 1)$ RSUs provide the average speeds $AvgSpd_{first}$, ..., $AvgSpd_{last}$, road conditions $RoadCond_{first}$, ..., $RoadCond_{last}$ together with signatures (U_{first}, W_{first}), ..., (U_{last}, W_{last}). Vehicle V_i verifies these signatures in a batch by checking whether $\hat{e}(\prod_{i=first}^{last} \sigma_i, g) = \prod_{i=first}^{last} \hat{e}(H(AvgSpd_i||RoadCond_i), RRID_i)$.

Proof of correctness:

$\hat{e}(\prod_{i=first}^{last} \sigma_i, g)$
$= \prod_{i=first}^{last} \hat{e}(\sigma_i, g)$
$= \prod_{i=first}^{last} \hat{e}(H(AvgSpd_i||RoadCond_i)^{RSK_i}, g)$
$= \prod_{i=first}^{last} \hat{e}(H(AvgSpd_i||RoadCond_i), g^{RSK_i})$
$= \prod_{i=first}^{last} \hat{e}(H(AvgSpd_i||RoadCond_i), RRID_i)$ □

We can see that with batch verification, vehicle V_i needs to perform only 2 pairing operations to verify the certificates of all RSUs. For the message verification, since the signatures are generated by different RSUs, altogether $(last - first + 2)$ pairing operations are needed.

X. Conclusions

We proposed a VANET-based Secure and Privacy-preserving Navigation (VSPN) scheme in this paper. We utilized speed data and road conditions collected by RSUs to guide vehicles to desired destinations in a distributed manner. Our scheme adopts some security primitives in a non-trivial way to provide a number of security features: 1) Vehicles are authenticated by means of pseudo identities. 2) Navigation queries and results are protected from eavesdroppers. Besides, with the idea of anonymous credential, no one including TA can link up a vehicle's navigation query and its identity. 3) Information provided by RSUs can be properly authenticated before the route is actually being used. Besides satisfying all security and privacy requirements, our solution is efficient in the sense that a vehicle can complete the whole navigation querying process and receive urgent notification in a very short time. On the other hand, the route returned by our scheme can lead to savings of up to 55% of travelling time compared with the offline map data searching approach. Our scheme also gives lower route blocking rate in practice. Note that our VSPN scheme can apply to the situation where the route searching process is done by a central server, which collects and verifies speed data and road conditions from RSUs. The authentication process at vehicles can be even simpler because a vehicle only needs to check against the central server's signature on the processed result. However, such a centralized approach is not scalable, especially for large cities. We are implementing our VSPN scheme on a testbed to further verify its performance.

Acknowledgment

Victor O.K. Li's research in this paper is supported in part by a Visiting Professorship at King Saud University, Saudi Arabia.

References

[1] "Global Positioning System Standard Positioning Service Signal Specification," book, 1995.

[2] "Papago! Z-Series Navigation System," 2009, http://www.papago.com.hk/.

[3] "Traffic Message Channel (TMC)," 2004, http://www.tmcforum.com/.

[4] F. Wang, D. Zeng, and L. Yang, "Smart Cars on Smart Roads: an IEEE Intelligent Transportation Systems Society Update," *IEEE Pervasive Computing, Vol. 5, No. 4*, pp. 68 – 69, 2006.

[5] H. Oh, C. Yae, D. Ahn, and H. Cho, "5.8 GHz DSRC Packet Communication System for ITS Services," in *Proceedings of the IEEE VTC '99*, Sept. 1999, pp. 2223 – 2227.

[6] I. Leontiadis, P. Costa, and C. Mascolo, "Extending Access Point Connectivity through Opportunistic Routing in Vehicular Networks," in *Proceedings of the IEEE INFOCOM '10*, Mar. 2010.

[7] C. Zhang, R. Lu, X. Lin, P. H. Ho, and X. Shen, "An Efficient Identity-based Batch Verification Scheme for Vehicular Sensor Networks," in *Proceedings of the IEEE INFOCOM '08*, Apr. 2008, pp. 816 – 824.

[8] R. Lu, X. Lin, H. Zhu, and X. Shen, "SPARK: A New VANET-based Smart Parking Scheme for Large Parking Lots," in *Proceedings of the IEEE INFOCOM '09*, Apr. 2009, pp. 1413 – 1421.

[9] D. Chaum, "Security without identification: Transaction systems to make Big Brother obsolete," *Communications of the ACM, Vol. 28*, pp. 1030 – 1044, 1985.

[10] E. Aimeur, H. Hage, and F. S. M. Onana, "Anonymous Credentials for Privacy-Preserving E-learning," in *Proceedings of the IEEE MCETECH '08*, July 2008, pp. 70 – 80.

[11] G. Samara, W.A.H. Al-Salihy, and R. Sures, "Security issues and challenges of Vehicular Ad Hoc Networks (VANET)," in *Proceedings of the IEEE NISS '10*, May 2010, pp. 393 – 398.

[12] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, "AMOEBA: Robust Location Privacy Scheme for VANET," *IEEE Journal on Selected Areas in Communications, Vol. 25, Issue 8*, pp. 1569 – 1589, 2007.

[13] C. Zhang, X. Lin, R. Lu, and P. H. Ho, "RAISE: An Efficient RSU-aided Message Authentication Scheme in Vehicular Communication Networks," in *Proceedings of the IEEE ICC '08*, May 2008, pp. 1451 – 1457.

[14] A. Wasef and X. Shen, "PPGCV: Privacy Preserving Group Communications Protocol for Vehicular Ad Hoc Networks," in *Proceedings of the IEEE ICC '08*, May 2008, pp. 1458 – 1463.

[15] "Researcher Cracks Trusted Platform Module Security Chip," http://www.digitaltrends.com/computing/researcher-cracks-trusted-platform-module-security-chip/.

[16] T.W. Chim, S.M. Yiu, L. C.K. Hui, and V. O.K. Li, "SPECS: Secure and Privacy Enhancing Communications for VANET," in *Proceedings of the ADHOCNETS '09*, Sept. 2009.

[17] B.K. Chaurasia, S. Verma, G.S. Tomar, and S.M. Bhaskar, "Pseudonym Based Mechanism for Sustaining Privacy in VANETs," in *Proceedings of the IEEE CICSYN '09*, Sept. 2009, pp. 420 – 425.

[18] R.J. Hwang, Y.K. Hsiao, and Y.F. Liu, "Secure Communication Scheme of VANET with Privacy Preserving," in *Proceedings of the IEEE ICPADS '11*, Dec. 2011, pp. 654 – 659.

[19] X. Lin, R. Lu, X. Liang, and X. Shen, "STAP: A social-tier-assisted packet forwarding protocol for achieving receiver-location privacy preservation in VANETs," in *Proceedings of the IEEE INFOCOM '11*, Apr. 2011, pp. 2147 – 2155.

[20] B. K. Chaurasia, S. Verma, and S. M. Bhasker, "Message broadcast in VANETs using Group Signature," in *Proceedings of the IEEE WCSN '09*, Dec. 2008, pp. 131 – 136.

[21] A. Studer, E. Shi, F. Bai, and A. Perrig, "TACKing Together Efficient Authentication, Revocation, and Privacy in VANETs," in *Proceedings of the IEEE SECON '09*, June 2009, pp. 1 – 9.

[22] J. P. Hubaux M. Raya, P. Papadimitratos, "Securing Vehicular Communications," *IEEE Wireless Communications, Vol. 13, Issue 5*, pp. 8 – 15, Oct. 2006.

[23] Y.S. Choi, J.T. Oh, J.S. Jang, and J.C. Ryou, "Integrated DDoS Attack Defense Infrastructure for Effective Attack Prevention," in *Proceedings of the IEEE ITCS '10*, Aug. 2010, pp. 1 – 6.

[24] A. Menezes, "An Introduction to Pairing-Based Cryptography," in *1991 Mathematics Subject Classification, Primary 94A60*, 1991.

[25] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-encryption Schemes with Applications to Secure Distributed Storage," in *Proceedings of the 12th Annual Network and Distributed Systems Security Symposium (NDSS)*, 2005.

[26] M. Green and G. Ateniese, "Identity-Based Proxy Re-encryption," in *Proceedings of the Applied Cryptography and Network Security Conference*, 2007.

[27] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," in *Proceedings of Asiacrypt '01*, 2001, pp. 514 – 532.

[28] D. Eastlake and P. Jones, "US Secure Hash Algorithm 1 (SHA1)," *IETF RFC3174*, 2001.

[29] D. Boneh, B. Lynn, and H. Shacham, "Short Signatures from the Weil Pairing," in *Proceedings of the ASIACRYPT '01, LNCS 2248, Springer-Verlag*, 2001, pp. 514 – 532.

[30] M. Scott, "Efficient implementation of cryptographic pairings," 2007, http://ecrypt-ss07.rhul.ac.uk/ Slides/Thursday/mscottsamos07.pdf.

[31] A. Miyaji, M. Nakabayashi, and S. Takano, "New explicit conditions of elliptic curve traces for FR-reduction," *IEICE Transactions on Fundamentals, Vol. E84-A, No. 5*, pp. 1234 – 1243, 2001.

[32] "Topologically Integrated Geographic Encoding and Referencing system (TIGER)," 2009, http://www.census.gov/geo/www/tiger/.

[33] "Google Map," http://maps.google.com.