Chapter 7

Privacy-Preserving Information Gathering Using VANET

T. W. Chim, S. M. Yiu, Lucas C. K. Hui and
Victor O. K. Li

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/53537

1. Introduction

For a driver, some real-time information (e.g. traffic condition along the road and the availability of parking spaces at certain areas) on his way to his destination may be useful. Nowadays, drivers could mainly rely on radio broadcasting. However, the traffic news on radio may not mention anything about the area you are driving into. A more effective way for providing this real-time information would be desirable.

In recent years, a special kind of ad hoc network called Vehicular Ad hoc NETwork (VANET) becomes increasingly popular. It has also become one of the critical components of an Intelligent Transportation Systems (ITS). 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 [1] 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. about vehicle speed, turning direction, road condition, 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. It is natural to consider whether VANET can provide an effective platform for drivers to utilize real-time information collected in RSUs.

In this chapter, we first highlight the most significant security and privacy challenges in VANET protocol design. We then discuss how one should design security protocols for VANETs. For example, we analyze in details the advantages and disadvantages of hardware-based and software-based solutions. Next we propose a VANET-based general information gathering scheme. A driver can issue a query (e.g. road conditions along the roads to his destination) to a nearby RSU, our scheme can then automatically collect the required information from the appropriate RSUs. The gathering process is done in a real-time and distributed manner. Note that the approach of using a centralized server that stores all information collected from RSUs may not work as the information may be changed frequently in a real-time manner and since the VANET is huge, the server will most likely become the bottleneck.

Like other existing VANET applications, there are basic security requirements to be satisfied by such a protocol. They include sender authentication (to ensure that the sender is a valid subscriber), conditional identity privacy preserving (to ensure that a driver's travelling route cannot be traced by any third party except the trusted authority). And there are additional security and privacy requirements to make it more practical. Conditional identity privacy preserving implies that a trusted authority is able to reveal the real identity of a vehicle. If the information gathering scheme is not properly designed, a driver's real identity and query (the information required) can be linked up and analyzed. This is not preferable because we want to ensure that no one in this world (including trusted authority) knows what a driver is querying for. This leads to our privacy preserving problem. Besides, a driver may not want vehicles nearby to know his query by eavesdropping his message. Also when the system sends the result back to him/her, we do not want non-subscribers nearby to enjoy free information gathering service in case it is a paid service. We regard this as a confidentiality problem. Finally, since our information gathering scheme involves the information provided by more than one RSU and RSUs are left unattended at roadsides most of the time, proper and efficient authentication of this information becomes critical. Our scheme addresses this authentication problem as well.

We provide a security analysis and a simulation study to evaluate our scheme. In our simulation, we make use of the maps of New York (city road system) and California (countryside highway system) downloaded from the TIGER database. We find that the processing time is at most 1.6 % and 3.7 %, respectively, of the duration that the vehicle stays in the querying RSU's range in the two cities. Thus there must be sufficient time for the vehicle to finish its query and to verify the returning information.

The rest of this chapter is organized as follows: Challenges of security protocols for VANETs are discussed in Section 2. Hardware and software approaches are then explained and compared in Section 3. The system model and the problem statement are described in Section 4. Some preliminaries on bilinear map are given in Section 5. Our schemes are presented in Section 6. The analysis and evalution of our schemes are given in Sections 7 and 8. Related work is reviewed in Section 9. Finally, Section 10 concludes the chapter.

2. Challenges of security protocols for VANETs

General security vulnerabilities and challenges for VANETs have been discussed in works like [2] and [3]. On the other hand, we focus on the challenges for designing security protocols in VANETs.

1. Dynamic, linear and real-time topology

   Moving vehicles are major components of VANETs. They are moving at high speed most of the time and this makes a VANET topology change rapidly and subject to frequent fragmentation. A vehicle which connects part of the VANET at a certain moment may no longer act as a connector in the next moment. Also, unlike MANET, nodes move in random direction, VANET vehicles move in a constrained manner. A vehicle must move along roads and change its direction only at junctions. Vehicles on a road tend to alight in a straight line. Security protocols for VANETs should not assume any fixed node infrastructure such as trees [4] [5]. Instead, dynamic topology should be properly handled. Furthermore, a VANET topology could be affected by drivers' reaction to messages. For example, a driver may change its route after receiving a message about congestion from another vehicle. Therefore, all tasks including those for security purpose should be performed in real-time. Centralized pre-processing is not possible.

2. Large scale and density varying network

   A VANET usually covers the whole region or even the whole city and thus the total number of VANET nodes can be very huge. This means that a centralized security protocol such as [6] may not be a good choice. Instead, operations have to be done in a decentralized and distributed manner. On the other hand, a VANET usually has different network density in different regions. For example, at where there is a traffic jam, the network becomes very dense. On the contrary, in suburban area, the network becomes very sparse. This implies when designing security protocols for VANETs, we cannot have the assumption of low or high network density. Instead, a good protocol should be able to handle both situations.

3. Transmission and computation efficiency

   One of the initial design goals of VANETs is the sharing of critical information (e.g. to inform vehicles about danger ahead of a road) among vehicles. Thus most messages in VANETs are of real-time importance. Therefore, security operations should cause as low overhead to the network as possible. In recent years, researchers start adopting elliptic curve cryptography (ECC) approach [7] to reduce key and ciphertext sizes. Also some other researchers are trying to reduce the computation overhead induced by security operations. For example, authors in [8] and [9] propose an efficient batch signature verification technique.

4. Conditional identity privacy perserving

   Normally, a driver may not want others to know his real identity and then trace his route or driving habit. If this cannot be satisfied, he may not subscribe to any new service including VANET at all. Thus the real identity of any vehicle should be kept anonymous from others and a third party should not be able to reveal a vehicleąęs real identity by analyzing multiple messages sent by it. RSUs are just installed along the roadside and are more vulnerable to attack. In the extreme case, even if all RSUs collude, we want to make sure that the relationship between the real identity of a vehicle and the messages it sent cannot be revealed.

   However, since vehicles are fast moving objects, injuries or even death are usually caused when accidents take place. If an accident is caused by a VANET message, a trusted party such as the police force may need to find out the message sender so as to avoid repeated occurrence of similar accidents. Thus while preserving a vehicleąęs privacy, its

real identity should be able to be traced by a trusted party when necessary. Thus we call the identity privacy preserving here conditional.

5. Small network diameter

   Because of the dynamic nature of VANETs, the network diameter, which is defined as the number of hops between the furthest endpoints of the network, can be very small. The network route between two vehicles may be disconnected easily due to moving out-of-range or having obstacle blocking. Thus those secure routing protocols originally designed for fixed Internet or MANETs cannot be directly adopted into VANETs. New protocols such as 'carry and forward' [10] have to be adopted to fit this specific property. Security issues induced by this kind of forwarding strategy are still open problems.

6. Multiple levels of security

   Different kinds of messages can exist in a nowadays VANET. As mentioned earlier, most messages are critical and are about conditions on the road. However, there can be others such as advertisements [11]. Thus a good security protocol should provide multiple security levels such as what is proposed by [12]. A critical message should have better protection than an advertisement message.

7. Energy efficiency considerations

   Unlike mobile ad hoc networks (MANETs) and wireless sensor networks (WSNs) where nodes are assumed to run on self batteries, energy is no longer a challenge in VANETs. It is because OBUs are continuously charged by car batteries while RSUs are continuously charged by fixed power cables. Thus researchers' attentions should be shifted back to security problems themselves rather than paid to energy efficiency directions.

3. Hareware and software approaches

Recent security works for VANETs go for a certificate-less direction (i.e. a sender does not need to send its certificate to the receiver for verification). However by nature, a vehicle's signature must contain a secret that is known by the receiver to facilitate the validation. To accomplish this, recent works focus on two major directions - hardware-based and software-based. In this section, we first explain how hardware-based and software-based solutions work respectively. Then we briefly discuss the advantages and disadvantages of each approach.

Hardware-based solutions usually rely on a tamper-proof device installed on a vehicle [3] [8]. It contains the secret we mentioned earlier and runs on its own battery and own clock. Thus an outsider cannot block its functions by cutting its power supply or by inputting wrong signals. Besides storing secrets, it is also in charge of all security operations including digital signature, encryption and decryption. Further, the device is accessible only by authorized personnel. A driver has to input a password before the device can function properly. A recent example of hardware-based solutions is [8]. Here the tamper-proof device installed on a vehicle has an accessing password preloaded. This password is assigned by a TA and is firmed burned onto the hardware when the device is first registered. Whenever a driver starts the vehicle, he/she has to input into the device the same accessing password in order to enable it for further operations. This is how the authentication of driver by the device is done. Besides the accessing password, the tamper-proof device also stores all system public

parameters together with the master keys s1 and s2 of the TA. These master keys are assumed to be known by only vehicles and the TA but not RSUs. They are used to form the signing keys for constructing signatures. TA's master keys also facilitates RSUs to verify vehicles' signatures even though they do not have any knowledge about the master keys. (It is based on pairing operations and interested readers please refer to [8] for details.) Note that the master keys can only be used for security operations inside the tamper-proof device. No outsiders, including the driver, know about their values.

Software-based solutions, on the contrary, do not rely on any tamper-proof device. What a vehicle has is an ordinary computer device. In other words, no secret or parameter can be preloaded securely onto a vehicle. However, it still requires them for security operations like digital signatures. Software-based solutions obtain these information through a secure initial handshaking. A recent software-based solution is presented in [9]. A conventional public key infrastructure is assumed to exist for initial secure message exchange. Whenever a driver starts the vehicle, he/she has to input into the computer device an accessing password. This password is pre-assigned by the TA and is assumed to be given to the driver earlier (e.g. via paper documents during car first registration). The password input by the driver is then encrypted using the TA's conventional public key which is assumed to be known by everyone. The TA decrypts the password using its conventional private key and checks whether it matches with its records. If yes, it encrypts its master keys using the vehicle's conventional public key. Upon receiving the encrypted message, the vehicle can obtain those master keys by decrypting the encrypted block using its conventional private key. A vehicle's conventional public and private key pairs are also assumed to be assigned by the TA at earlier stage (e.g. during car first registration). Thus the computer device can perform security operations as what a tamper-proof device does. Based on the above brief descriptions, we can see that both hardware-based and software-based solutions can resolve the challenges we mentioned in Section 2:

1. They do not have any assumptions on the network nature and so they can fit the dynamic, linear, density-varying and small-diameter VANET topology well.
2. Except initial handshaking, all cryptographic operations are done in real-time and so they are suitable for real-time-changing VANET environments.
3. Both of them achieve transmission and computation efficiency. First, they adopt ECC which possess the property of short key. Second, efficient signature batch verification routines were proposed (please refer to [8] and [9] for details). Third, unlike traditional public key infrastructure, a vehicle does not need to send its certificate to others for signature verification purpose. Finally, unlike mobile phone network which is mainly for unicast, Dedicated Short Range Communications (DSRC) [1], which are short to medium range wireless communications channels specifically designed for automative use, can facilitate efficient broadcast.
4. A vehicle only attaches a pseudo identity in its messages. Its real identity can only be traced by TA using a tracing routine (please refer to [8] and [9] for details). Thus conditional identity privacy can be preserved.
5. Regarding multiple levels of security, there is already a representative work [12]. Thus, the extension is not difficult and is practical actually.

Next let us compare the two approaches from a number of aspects.

1. Authentications of drivers

   For hardware-based solutions, to authenticate a driver, a tamper-proof device only needs to check the accessing password input by the driver locally. If it does not match the burned one, it simply disables all its functions. However, for software-based solutions, the ordinary computer device does not know whether the accessing password input by the driver is correct or not because it has no secret pre-stored. Hence, it needs to securely transmit it via network to the TA for further verification.

2. System parameters preloading

   For hardware-based solutions, the system-wide TAąęs master keys are preloaded into a tamper-proof device. Thus, the TA does not need to send them to the device anymore after an initial hardware burning. However, for software-based solutions, no secret or parameter is stored in the ordinary computer device. Thus every time a driver starts the vehicle and after accessing password checking, the TA has to send them again to the device. Extra transmission overhead is needed in each session.

3. Replication of device contents

   One basic assumption of tamper-proof device or smart card technology is that the contents inside the device or smart card cannot be improperly extracted or replicated easily. Thus secrets and parameters stored on them can be said to be fully protected. However, the case is not the same for ordinary computer device in software-based solutions. As everyone knows, the contents of a hard-disk can be cloned or replicated easily. This is why in the software-based solutions mentioned earlier, system secrets cannot be stored in a computer device. Instead they have to be transmitted from the TA every time it starts up.

4. Updates of system parameters

   For hardware-based solutions, the system-wide TAąęs master keys and other public parameters cannot be updated easily. Once an update is needed, a driver has to physically bring the device to the TA for an update. However, for software-based solutions, all secrets and public parameters are transmitted from the TA in real time when the vehicle starts up. Thus updates can be easily done. All the TA needs to do is to send the new set of secrets and parameters to the device.

5. Setting of new secrets after compromise

   For hardware-based solutions, the same set of system-wide TAąęs master keys and public parameters are preloaded into all tamper-proof devices. Once one of the devices is cracked by an attacker, the whole VANET system will be compromised unless a physical hardware update by the TA is done. However, software-based solutions do not have this problem. When one of the devices is found to be cracked, the TA can invoke an update of master keys and parameters by simple and secure network transmissions.

6. Modification of protocols

   For hardware-based solutions, all security operations are carried out by the processor of a tamper-proof device. Thus the same set of security operations are preloaded into all devices. If, in the mean time, the TA wants to introduce a new security operation (e.g. to enhance the security level of the VANET system), it needs to ask all drivers to bring the devices to it for a hardware update. However, software-based solutions do not have this problem. When the TA wants to introduce a new security operation, it only needs to enable all computer devices to securely download a new software (like how we update our computer operating systems nowadays).

Features	Hardware-based	Software-based
Authentications of drivers	Local tamper-proof device only	Transmission to TA needed
System-wide secrets and public parameters preloaded	Yes	No
Replication of device contents	Very difficult	Relatively easier
Updates of system-wide secrets and public parameters	Physical hardware update at TA	Simple secure download
Setting of new secrets after compromise	Physical hardware update at TA	Simple secure download
Modification of protocols	Physical hardware update at TA	Simple secure download
Complexity of security operations	Need to be simple	Relatively more complicated

Table 1. Hareware-based vs. Software-based Solutions

7. Complexity of security operations

   For hardware-based solutions, all security operations are carried out by the processor of a tamper-proof device. Unluckily, the computation power of the processor is quite limited. Up to our knowledge, not all smart cards in the market today are powerful enough to perform pairing operations. Thus, security operations adopted have to be as simple as possible. However, software-based solutions do not have this limitation. Even the poorest CPU today can handle complicated operations like pairing in reasonable time.

We summarize the comparisons between hardware-based, hybrid and software-based solutions in Table 1. We can see that the items are around efficiency, flexibility and security. In short, recall that a VANET is of large scale, hardware-based solutions are more efficient than software-based ones since they can reduce the transmission overhead between devices and TA during initial handshaking. However, there may still be problems when updating of system parameters, secrets or cryptographic protocols is required since if all drivers go to the TA for parameters, hardware or software updates, a bottleneck will appear. Our proposed scheme is hardware-based but we also provide suggestions about how to update system parameters, secrets or cryptographic protocols efficiently.

4. System model and assumptions

Besides the assumptions made in other VANET applications such as TA being trusted and real identity of any vehicle being known by TA and itself but not by others, we further assume the followings:

1) There exists a conventional identity-based public key infrastructure (PKI). 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 the RSU-Vehicle Communications (RVC) range of it. The validity of $RRID_i$ can be ensured using a certificate issued by the TA.

2) Each RSU has a local database storing road information in its range (e.g. GPS locations of boundaries, names of buildings and streets, etc.). This facilitates an RSU to answer queries that are about fixed facilities in its range.

3) Each vehicle has a tamper-proof device and a conventional computer device with GPS receiver. The tamper-proof device is responsible for generating pseudo identities and signatures on messages (details will be given later in the next section) and is assumed to have its own clock for generating correct time stamps and be able to run on its own battery [3]. The conventional computer device is responsible for all other calculations and can receive GPS signals.

4) We assume that there is a reasonably large number of information gathering queries issued to RSUs. Otherwise, if there is only one query, the sender can be linked up with the query easily.

5. Preliminaries

Our security scheme is *pairing-based* and defined on two cyclic groups with a mapping called *bilinear map* [7]. In this section, we briefly introduce what a bilinear map is.

Let $\mathbb{G}$ be a cyclic additive group and $\mathbb{G}_{\mathbb{T}}$ be a cyclic multiplicative group. 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: $\forall P, Q, R \in \mathbb{G}$ and $\forall a, b \in \mathbb{Z}$, $\hat{e}(Q, P + R) = \hat{e}(P + R, Q) = \hat{e}(P, Q) \cdot \hat{e}(R, Q)$. Also $\hat{e}(aP, bP) = \hat{e}(P, bP)^a = \hat{e}(aP, P)^b = \hat{e}(P, P)^{ab}$.
2. Non-degenerate: There exists $P, Q \in \mathbb{G}$ such that $\hat{e}(P, Q) \neq 1_{\mathbb{G}_{\mathbb{T}}}$.
3. Computable: There exists an efficient algorithm to compute $\hat{e}(P, Q)$ for any $P, Q \in \mathbb{G}$.

The bilinear map $\hat{e}$ can be constructed using pairings on elliptic curves. Each operation for computing $\hat{e}(P, Q)$ is referred 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 $Q = aP$, there exists no efficient algorithm to obtain a by given P and Q. The implication is that we can transfer Q in an open wireless channel without worrying that a (usually some secret) can be known by any attackers.

6. Our scheme

This section presents our Privacy-preserving Information Gathering scheme. We first summarize our scheme into some basic steps (see Figure 1):

1) TA sets up parameters and generates anonymous credentials.

2) Vehicle V_i requests for a credential from RSU R_j.

3) RSU R_j verifies V_i's identity and sends it a credential.

4) After a random delay or after travelling for a random distance, V_i sends out its request to RSU R_k.

http://dx.doi.org/10.5772/53537

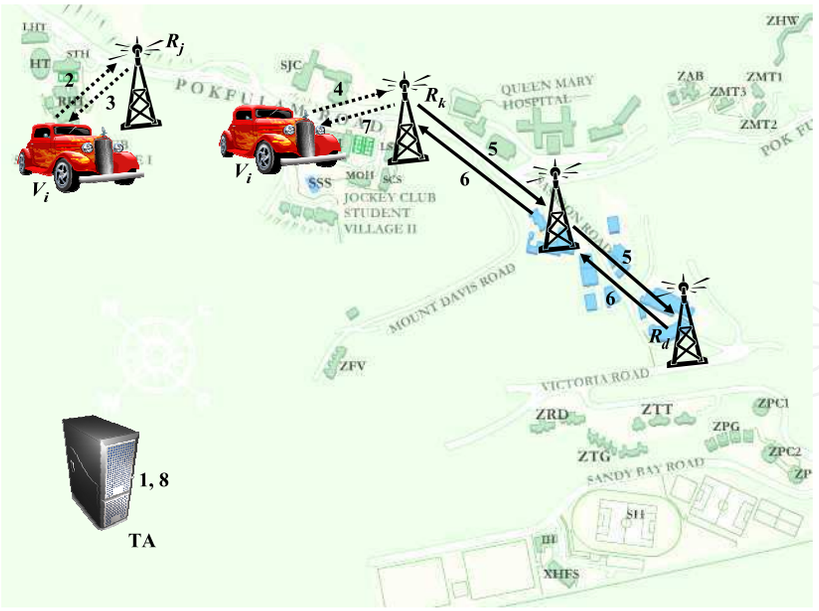

Figure 1. Basic Steps in Our Scheme

5) RSU R_k forwards the request to its neighbors. This process repeats until the request reaches the RSU covering the furthest point of interest with respect to V_i's current location.

6) RSU R_d constructs the information reply message and sends it along the reverse path. Each hop whose range overlaps with the region of interest attaches the corresponding hop information (with signature).

7) RSU R_k forwards the reply message to V_i which then verifies the messages from all RSUs along the route in a batch.

8) 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 chapter are summarized in Table 2.

6.1. Setup

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

1. It chooses groups $\mathbb{G}$ (with g as the generator) and $\mathbb{G}_{\mathbb{T}}$ that satisfy bilinear map properties.
2. It randomly picks $s \in \mathbb{Z}_q$ as the master secret (preloaded into all vehicles' tamper-proof devices).
3. It computes $g_{pub} = g^s$ as a public parameter. Note that given $g_{pub} = g^s$, there exists no efficient algorithm to obtain s based on the fact that the discrete logarithm problem (DLP) on bilinear groups is computationally hard.
4. It assigns itself a secret key TSK and an identity $TRID = g^{TSK}$ which is assumed to be known by everyone in the system.

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
$TRID$	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
$VRID_i$	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 [13] on message M
$h(M)$	One-way hash value of message M

Table 2. Notations used in this chapter

5. It assigns each RSU R_i locating at RL_i a secret key RSK_i, an identity $RRID_i = g^{RSK_i}$ and generates its certificate as $RC_i =< RRID_i, RL_i, TSIG_{TSK}(RRID_i||RL_i) >$ where $TSIG_{TSK}(RRID_i||RL_i) = H(RRID_i||RL_i)^{TSK}$ is TA's signature on the concatenation of $RRID_i$ and RL_i. Here $H(.)$ is a MapToPoint hash function.
6. It assigns each vehicle V_i a real identity $VRID_i = g^x$ where x is a random number and can be thrown away after generating $VRID_i$, and the hardware activation password $VPWD_i$. TA preloads them into the tamper-proof device of V_i.
7. It assigns each vehicle V_i a pair of conventional public key $VCPK_i$ and private key $VCSK_i$ under any public key infrastructure. $VCSK_i$ is preloaded into the tamper-proof device of V_i while $VCPK_i$ is stored into TA's local database. This conventional public and private keys are for updating the master secret s when there is a need (e.g. when any vehicle is proved to be compromised and the master secret is leaked to attackers). During such an update, TA can encrypt and send the new master secret to each uncompromised vehicle V_i using the corresponding $VCPK_i$. In this way, only the uncompromised vehicles can decrypt and obtain the new master secret.

http://dx.doi.org/10.5772/53537

Throughout this chapter, let us use the notations $AS\_ENC_x(M)$ and $S\_ENC_x(M)$ to denote encrypting message M using the key x based on any asymmetric and symmetric encryption algorithms, respectively.

6.2. Generation of anonymous credentials by TA

In our scheme, a credential will expire after a predefined period of time. Thus even if a subscriber leaks its credential to a non-subscriber or even to an attacker, the impact to the system is limited. Assume that the current time is T. TA computes the credential for the current period as $C_T =< \mathbf{CRD}, T, TSIG_{TSK}(\mathbf{CRD}||T) >$, where $TSIG_{TSK}(\mathbf{CRD}||T) = H(\mathbf{CRD}||T)^{TSK}$, and sends it to all RSUs securely via a fixed infrastructure. We can see that the credential carries no information about any user and that is why we call it "anonymous".

6.3. Activation of tamper-proof device on vehicle V_i

When the vehicle V_i starts, the driver enters the real identity $VRID_i$ and password $VPWD_i$ (assigned by TA in Section 6.1) into the tamper-proof device to activate it. Here only simple hardware checking is involved. The tamper-proof device continues with its pseudo identity generation and message signing tasks only if both the real identity and the password are correct. That means V_i cannot use the service if it is being stolen.

6.4. Vehicle V_i requesting for anonymous credential at RSU R_j

To request for an anonymous credential, V_i's tamper-proof device performs the following steps:

1. It generates a pseudo identity $VPID_i = (VPID_{i1}, VPID_{i2}) = (g^r, VRID_i \oplus H(g^r_{pub}))$ where r is a per-session random nonce.
2. It composes the credential request message $M_i = \{\mathbf{CRD\_REQ}\}$.
3. It 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.
4. It generates the signing key $VSK_i = (VSK_{i1}, VSK_{i2}) = (VPID^s_{i1}, HP^s_i)$ where $HP_i = H(VPID_{i1}||VPID_{i2})$.
5. It generates the 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.
6. It sends $< AS\_ENC_{RRID_j}(rand), VPID_i, M_i, T_i, \sigma_i >$ to RSU R_j nearby.

The RSU R_j then performs the following steps:

1. It 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 reply attack.

2. It verifies V_i's signature by checking whether $\hat{e}(\sigma_i, g) = \hat{e}(VPID_{i1} \times HP_i^{h(M_i||T_i)}, g_{pub})$.
   Proof of correctness:
   L.H.S.
   $= \hat{e}(\sigma_i, g)$
   $= \hat{e}(VSK_{i1} \times VSK_{i2}^{h(M_i||T_i)}, g)$
   $= \hat{e}(VPID_{i1}^s \times HP_i^{sh(M_i||T_i)}, g)$
   $= \hat{e}(VPID_{i1} \times HP_i^{h(M_i||T_i)}, g^s)$
   $= \hat{e}(VPID_{i1} \times HP_i^{h(M_i||T_i)}, g_{pub})$
   = R.H.S. □
3. If it receives requests from more than one vehicle at the same time (say request messages M_{first}, ..., M_{last}, signatures σ_{first}, ..., σ_{last} from vehicles V_{first}, ..., V_{last} respectively), it verifies them in a batch by checking whether $\hat{e}(\prod_{i=first}^{last} \sigma_i, g) = \hat{e}(\prod_{i=first}^{last} VPID_{i1} \times HP_i^{h(M_i||T_i)}, g_{pub})$.
   Proof of correctness:
   L.H.S.
   $= \hat{e}(\prod_{i=first}^{last} \sigma_i, g)$
   $= \hat{e}(\prod_{i=first}^{last} VSK_{i1} \times VSK_{i2}^{h(M_i||T_i)}, g)$
   $= \hat{e}(\prod_{i=first}^{last} VPID_{i1}^s \times HP_i^{sh(M_i||T_i)}, g)$
   $= \hat{e}(\prod_{i=first}^{last} VPID_{i1} \times HP_i^{h(M_i||T_i)}, g^s)$
   $= \hat{e}(\prod_{i=first}^{last} VPID_{i1} \times HP_i^{h(M_i||T_i)}, g_{pub})$
   = R.H.S. □
4. For each vehicle whose signature is valid, R_j encrypts the anonymous credential for the current period C_T using *rand* and sends $S\_ENC_{rand}(C_T)$ back to it.

6.5. Vehicle V_i requesting for information at RSU R_k

Note that if V_i obtains the credential C_T from RSU R_j and if it sends out its query to R_j immediately, its real identity and its query can always be linked up once R_j colludes with TA. Thus we propose two approaches to avoid this from happening:

1. V_i sends out its 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.
2. V_i sends out its 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 assume that V_i sends its query to RSU R_k. V_i performs the following:

1. It composes the request message $M_i = \{\mathbf{SREQ}, LOC_i, Interest_i\}$ where LOC_i represents the current location of V_i and $Interest_i$ contains a set of points of interest (in GPS coordinates) and description of information required (e.g. average vehicle speed, congestion status).

2. It picks two random numbers *rand* and *sn*. *rand* is for R_k to encrypt the result at a later stage and *sn* is used as a session number.
3. It sends $< AS\_ENC_{RRID_k}(rand, sn, C_T, M_i) >$ to R_k and stores *rand* and *sn* locally.

R_k then performs the following steps:

1. It decrypts the message using its private key.
2. It ensures the credential used C_T is not outdated (e.g. the timestamp should be within a pre-defined number of periods before the current time).
3. It verifies TA's signature on C_T by checking whether $\hat{e}(TSIG_{TSK}(\mathbf{CRD}||T), g) = \hat{e}(H(\mathbf{CRD}||T), TRID)$.
   Proof of correctness:
   L.H.S.
   $= \hat{e}(TSIG_{TSK}(\mathbf{CRD}||T), g)$
   $= \hat{e}(H(\mathbf{CRD}||T)^{TSK}, g)$
   $= \hat{e}(H(\mathbf{CRD}||T), g^{TSK})$
   $= \hat{e}(H(\mathbf{CRD}||T), TRID)$
   = R.H.S. □
4. It the signature is valid, it proceeds to the information gathering process.
5. It stores *rand* and *sn* locally for later usage.

6.6. Request and reply propagation

RSU R_k takes up the role of initiating the information gathering process by composing the information request message $M_k = \{\mathbf{INFO\_REQ}, sn, RRID_k, LOC_i, Interest_i\}$. Let FP be the furthest point with respect to LOC_i in $Interest_i$. R_k broadcasts M_k to all neighbors which are closer to FP than itself.

Any receiving RSU first stores sn, $RRID_k$ and $Interest_i$ into its routing table to build up the reverse path so that it can send any reply back to R_k later on. Let FP be the furthest point with respect to LOC_i in $Interest_i$. It then checks whether FP is within its range. If not, it simply re-broadcasts M_k to all neighbors which are closer to FP than itself. Otherwise, it computes the information reply message $M_d = \{\mathbf{INFO\_RPY}, sn, RRID_d, RL_d, RC_d, HopInfo_d, \sigma_d\}$ and sends it back to its previous RSU hop. Here $HopInfo_d$ is the information that is of V_i's interest and $\sigma_d = H(HopInfo_d)^{RSK_d}$ is R_d's signature on $HopInfo_d$.

Each RSU hop along the reverse path R_{im} repeats the steps done by R_d and if any point in $Interest_i$ is within its range, it includes information and signature corresponding to its hop (i.e. $HopInfo_{im}$ and σ_{im}) into the information reply message. Otherwise, it simply forwards the reply message to its previous RSU hop.

Upon receiving a reply, R_k encrypts it using *rand* and forwards it to V_i immediately.

6.7. Verification of RSUs' hop information

Recall that vehicle V_i receives from R_k a set of identities, a set of hop information and a set of signatures, each corresponding to an RSU along the path of propagation. To verify the hop information 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.

Let us first talk about how the RSUs' certificates can be verified in a batch. 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:
L.H.S.
$= \hat{e}(\prod_{i=first}^{last} TSIG_{TSK}(RRID_i||RL_i), g)$
$= \hat{e}(\prod_{i=first}^{last} H(RRID_i||RL_i)^{TSK}, g)$
$= \hat{e}((\prod_{i=first}^{last} H(RRID_i||RL_i))^{TSK}, g)$
$= \hat{e}(\prod_{i=first}^{last} H(RRID_i||RL_i), g^{TSK})$
$= \hat{e}(\prod_{i=first}^{last} H(RRID_i||RL_i), TRID)$
= R.H.S. □

Further assume these $(last - first + 1)$ RSUs provide the hop information $HopInfo_{first}$, ..., $HopInfo_{last}$ together with signatures $(\sigma_{first}, ..., \sigma_{last}$. Vehicle V_i verifies these signatures in a batch by checking whether $\hat{e}(\prod_{i=first}^{last} \sigma_i, g) = \prod_{i=first}^{last} \hat{e}(H(HopInfo_i), RRID_i)$.

Proof of correctness:
L.H.S.
$= \hat{e}(\prod_{i=first}^{last} \sigma_i, g)$
$= \prod_{i=first}^{last} \hat{e}(\sigma_i, g)$
$= \prod_{i=first}^{last} \hat{e}(H(AvgSpd_i||RoadCond_i)^{RSK_i}, g)$
$= \prod_{i=first}^{last} \hat{e}(H(AvgSpd_i||RoadCond_i), g^{RSK_i})$
$= \prod_{i=first}^{last} \hat{e}(H(AvgSpd_i||RoadCond_i), RRID_i)$
= R.H.S. □

We can see that 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. Note that the above verification procedures still apply even if the returned route contains only one single hop R_k. In that case, we can simply set $first = last = k$ in the expressions.

6.8. Traceability of vehicle V_i's real identity

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

7. Security analysis

We analyze our scheme with respect to the security and privacy requirements mentioned earlier.

1) Conditional identity privacy preserving: The pseudo identity of any vehicle is an ElGamal-type ciphertext, which is secure under the chosen plaintext attacks [14]. Also the random nonce r makes them different in different messages. 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. On the other hand, Section 6.8 shows that TA is able to trace a vehicle's real identity. Thus no one except TA can trace the real identity of a particular vehicle and conditional identity privacy is achieved.

2) Privacy preserving and unlinkability: After vehicle V_i obtains an anonymous credential, it presents it to the same RSU after a random delay or to a different RSU for service as discussed earlier. In any case, that RSU does not know V_i's pseudo identity and identity verification is based on an anonymous credential, it cannot link up V_i's query with its identity even if it colludes with TA. Thus unlinkability is guaranteed.

3) Confidentiality: When vehicle V_i requests for a 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 credential using *rand*. Thus neighboring vehicles cannot illegally receive the credential by eavesdropping messages from the air. Similarly, when vehicle V_i requests for information gathering service from RSU R_k, it picks another random number and R_k in return encrypts the result using that random number. Thus no other vehicles can eavesdrop the result even if they are interested in similar information. For the query, V_i encrypts it using RSU's identity and so it is kept confidential from others.

4) Message authentication: 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.

Regarding vehicle V_i's signature, it is composed of VSK_{i1} and VSK_{i2}. VSK_{i1} is defined as g^{rs}. Due to the difficulty of solving the discrete logarithm problem, there is no way for attackers to obtain s and thus no one other than the tamper-proof device can compose VSK_{i1}. VSK_{i2}, on the other hand, is defined as HP_i^s. Again, since no one other than tamper-proof devices knows s, VSK_{i2} cannot be forged as well.

8. Simulation results

In this section, we evaluate our scheme in terms of processing delay using a network simulation program. Through simulation, we show that the processing delay caused by our cryptographic functions is minimal.

8.1. Simulation models

In our simulation, we made use of two maps downloaded from the TIGER database [15] - one is New York and the other is California. New York represents a city road system (see Fig. 2 for the Google Map [16]) 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 [16]) in which some highways have speed limit up to 120 km/h. RSUs are randomly placed onto each

road. 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).

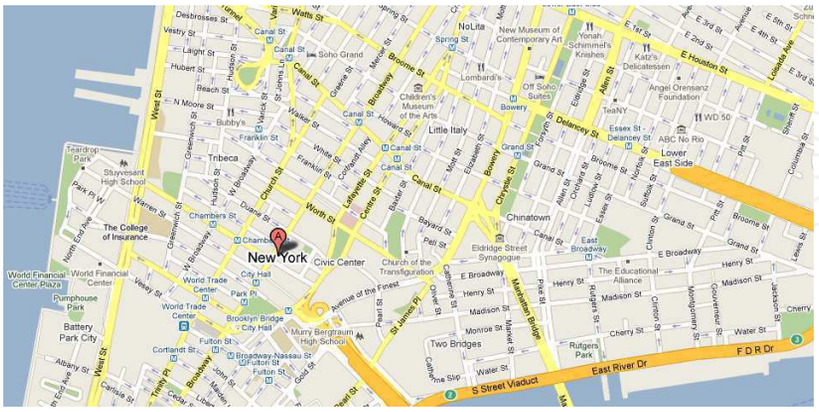

Figure 2. City Road System in New York

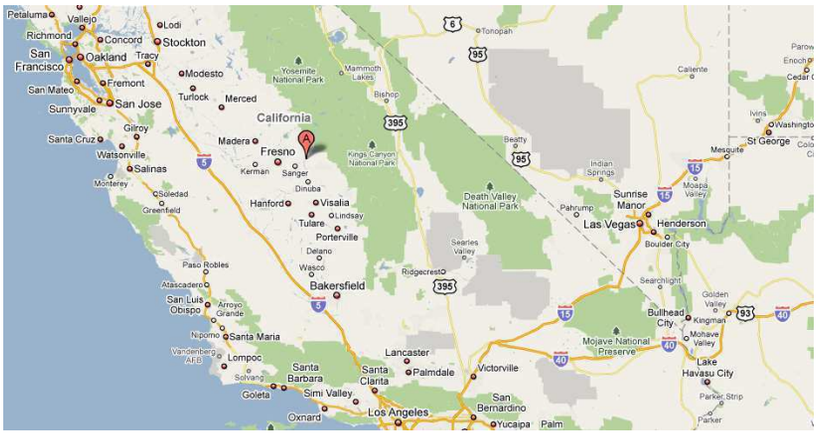

Figure 3. Countryside Highway System in California

The settings and parameters of our simulation are adopted from [8] and [9]. Interested readers may refer to them for details. We fix the size of our newly-introduced components as follows: 5 bytes for control messages like **CRD_REQ**, 20 bytes for each representation of GPS location, 255 bytes for timestamp and 10 bytes for random number.

We define 16 geographical distance ranges of 1 km each. For New York, the closest source and destination we pick are only 1 km apart while the furthest are 16 km. For California, the closest source and destination we pick are only 5 km apart while the furthest are 80 km. 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 furthest point of interest of a querying vehicle respectively. We then consider the worst case that all points between the current location and the furthest point of interest are of the driver's interest. The types of information we consider are average vehicle speed and general road condition (e.g. accident, traffic jam). Without loss of generality, we assume that the vehicle requests for a credential or sends out its 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 credential before sending out its query, we define the processing

http://dx.doi.org/10.5772/53537

time as the period from when the vehicle sends out its query to when it finishes verifying the information provided by all RSUs in the reply message. This processing time is then normalized by dividing it by the duration that the vehicle is in the range of the RSU to which it sends its query. The data from all the 60 sets are then averaged to obtain a data point as shown in Figure 4 below. Note also that we represent a range using its class mark.

8.2. Simulation results

Fig. 4 shows the results for New York city. We can see that as the geographical distance increases, the processing time increases. When the source and the destination nodes are further away (i.e. a vehicle wants to gather information about a point of interest which is 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. Nevetheless, among all geographical distance ranges, the processing time is at most 1.6 % of the duration that the vehicle stays in the querying RSU's range. Thus there must be sufficient time for the vehicle to finish its query and to verify the returning information.

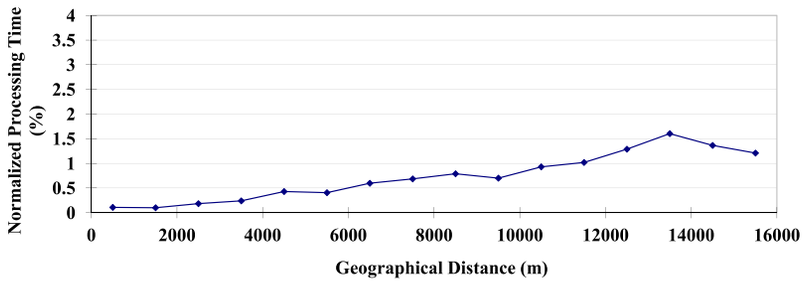

Figure 4. Normalized Processing Time vs. Geographical Distance (New York)

Fig. 5 shows the results for California city. We can see again that as the geographical distance increases, the processing time increases. When the source and the destination nodes are further away (i.e. a vehicle wants to gather information about a point of interest which is 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. Among all geographical distance ranges, the processing time is at most 3.7 % of the duration that the vehicle stays in the querying RSU's range. This value is a little bit greater than that for New York city due to larger geographical distances. Anyway, there must be sufficient time for the vehicle to finish its query and to verify the returning information.

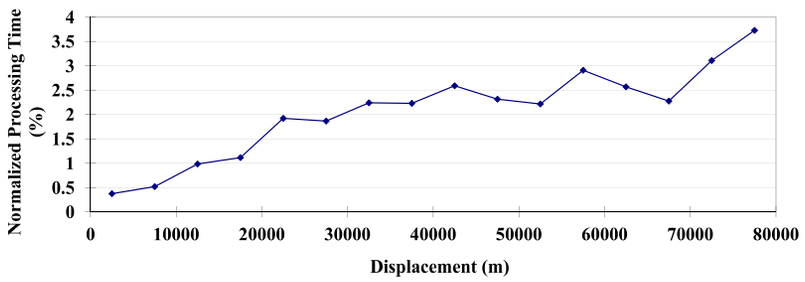

Figure 5. Normalized Processing Time vs. Geographical Distance (California)

9. Related work

A similar scheme of real-time information gathering using VANET is proposed in a recent work [17]. However, there are a number of differences between their scheme and ours. First, their scheme is a small scale navigation scheme 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 centrally 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. That is, all information are provided by the three RSUs. In our scheme, the road system in the city is monitored by a large number of RSUs which take up the information gathering 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 indoor and can be monitored by security guards from time to time. However, such an assumption is no longer valid in our outdoor setting. It is impossible to have security guards monitoring 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 query to be delinked. This feature is only interesting for wide area information gathering like ours. Thus, the scheme provided in [17] cannot be used to solve the information gathering problem discussed in this chapter.

Other recent efforts related to the security issues in VANET include [8, 9, 18–21]. In [8], a batch verification scheme was proposed for an RSU to verify a large number of signatures at the same time using only three pairing operations. In [18], an RSU-aided inter-vehicle communications scheme was proposed. A vehicle relies on an RSU to verify the signature of another vehicle. In [19], group communications in VANETs are considered and a group key update protocol was proposed. In [9], some security and privacy-enhancing communications schemes were proposed. 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. [20] and [21] 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, group signature schemes are rather complicated and may not be practical.

10. Conclusions

In this chapter, we first highlighted the most significant security and privacy challenges in VANET protocol design. We then discussed how one should design security protocols for VANETs. In particular, we analyzed in details the advantages and disadvantages of hardware-based and software-based solutions. Next we proposed an information gathering scheme using VANETs. We utilized information collected by RSUs to provide drivers information about a set of points of interest that is out of sight in a distributed manner. Besides basic security features such as sender authentication and conditional identity privacy preserving. Our scheme adopts some security primitives in a non-trivial way to provide a number of additional security features: 1) With the idea of anonymous credential, no one including TA can link up a vehicle's query and its identity. 2) Queries and resulting information are protected from eavesdroppers. 3) Information provided by RSUs are

properly authenticated. We provided a security analysis and a simulation study to evaluate our scheme. In our simulation, we made use of the maps of New York (city road system) and California (countryside highway system) downloaded from the TIGER database. We found that the processing time is at most 1.6 % and 3.7 %, respectively, of the duration that the vehicle stays in the querying RSU's range in the two cities. Thus there must be sufficient time for the vehicle to finish its query and to verify the returning information.

Author details

T. W. Chim[1,2,⋆], S. M. Yiu[1], Lucas C. K. Hui[1] and Victor O. K. Li[2]

⋆ Address all correspondence to: twchim@cs.hku.hk; smyiu@cs.hku.hk; hui@cs.hku.hk; vli@eee.hku.hk

1 Department of Computer Science, The University of Hong Kong, Hong Kong
2 Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong

References

[1] 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*, pages 2223 – 2227, September 1999.

[2] J.P. Hubaux, S. Capkun, and J. Lui. The Security and Privacy of Smart Vehicles. *IEEE Security and Privacy Magazine, 2(3)*, pages 49 – 55, 2004.

[3] J. P. Hubaux M. Raya, P. Papadimitratos. Securing Vehicular Communications. *IEEE Wireless Communications, Vol. 13, Issue 5*, pages 8 – 15, October 2006.

[4] Y. Kim, A. Perrig, and G. Tsudik. Tree-Based Group Key Agreement. *ACM Transactions on Information Systems Security, 7(1)*, pages 60 – 96, 2004.

[5] Y. Jiang, M. Shi, X. Shen, and C. Lin. A Tree-Based Signature Scheme for VANETs. In *Proceedings of the IEEE GLOBECOM*, pages 1 – 5, 2008.

[6] C.K. Wong, M. Gouda, and S.S. Lam. Secure Group Communications using Key Graphs. In *Proceedings of the IEEE SIGCOMM*, pages 68 – 79, 1998.

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

[8] 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*, pages 816 – 824, April 2008.

[9] T.W. Chim, S.M. Yiu, Lucas C.K. Hui, and Victor O.K. Li. SPECS: Secure and Privacy Enhancing Communications for VANET. *Elsevier Ad Hoc Networks, Vol. 9, Issue 2*, pages 189 – 203, March 2010.

[10] J. Jakubiak and Y. Koucheryavy. State of the Art and Research Challenges for VANETs. In *Proceedings of the IEEE CCNC*, pages 912 – 916, 2008.

[11] S.B. Lee, G. Pan, J.S. Park, M. Gerla, and S. Lu. Secure Incentives for Commercial Ad Dissemination in Vehicular Networks. In *Proceedings of the ACM MobiHoc*, pages 150 – 159, 2007.

[12] T.W. Chim, S.M. Yiu, Lucas C.K. Hui, and Victor O.K. Li. MLAS: Multiple Level Authentication Scheme for VANETs. *Elsevier Ad Hoc Networks Journal, Vol. 10, Issue 7*, September 2012.

[13] D. Boneh, B. Lynn, and H. Shacham. Short Signatures from the Weil Pairing. In *Proceedings of Asiacrypt '01*, pages 514 – 532, 2001.

[14] J. Baek, B. Lee, and K. Kim. Secure Length-Saving ElGamal Encryption under the Computational Diffie-Hellman Assumption. *Lecture Notes in Computer Science - Information Security and Privacy, Vol. 1841*, pages 49 – 58, 2000.

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

[16] Google Map. http://maps.google.com.

[17] 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*, pages 1413 – 1421, April 2009.

[18] 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*, pages 1451 – 1457, May 2008.

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

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

[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*, pages 1 – 9, June 2009.