<table>
<thead>
<tr>
<th><strong>Title</strong></th>
<th>OPQ: OT-based private querying in VANETs</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Author(s)</strong></td>
<td>Chim, TW; Yiu, SM; Hui, LCK; Li, VOK</td>
</tr>
<tr>
<td><strong>Citation</strong></td>
<td>IEEE Transactions on Intelligent Transportation Systems, 2011, v. 12 n. 4, p. 1413-1422</td>
</tr>
<tr>
<td><strong>Issued Date</strong></td>
<td>2011</td>
</tr>
<tr>
<td><strong>URL</strong></td>
<td><a href="http://hdl.handle.net/10722/140798">http://hdl.handle.net/10722/140798</a></td>
</tr>
<tr>
<td><strong>Rights</strong></td>
<td>©2011 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.</td>
</tr>
</tbody>
</table>
OPQ: OT-Based Private Querying in VANETs

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

Abstract—We consider the querying service (e.g., location-based query service) in vehicular ad hoc networks (VANETs). Querying service has been studied in various kinds of networks such as traditional mobile phone networks and other mobile ad hoc networks. However, existing schemes are either not suitable for VANETs due to their highly dynamic environment or do not provide a privacy-preserving solution. In this paper, we first discuss the security concerns of providing a querying service that ensures that a query will not be linkable to the querier. Then, we briefly highlight the characteristics of VANETs, which make the problem different from other types of networks. Finally, we propose a solution for solving the problem by using techniques of pseudoidentity, indistinguishable credentials, and oblivious transfer. We show that, although all infrastructure units collude, it is still impossible to link the real identity of the user to a query. Based on our simulation study, we show that our scheme is effective in terms of processing delay, message overhead, and success rate.

Index Terms—Oblivious transfer (OT), privacy preserving, pseudoidentity, querying service, vehicular sensor network.

I. INTRODUCTION

A VEHICULAR ad hoc network (VANET) is initially designed for enhancing driving safety and convenience through intervehicle communications (IVCs) or communications with units in the roadside infrastructure. It is an important element of the intelligent transportation systems (ITSs) [1]. In a typical VANET, each vehicle has an on-board unit (OBU), and along the roads, road-side units (RSUs) are installed. A trusted authority (TA) and, maybe, some other application servers are installed in the backbone. OBUs and RSUs communicate using the Dedicated Short-Range Communications (DSRC) protocol [2] over the wireless channel, whereas RSUs, the TA, and the application servers communicate using a secure fixed network (e.g., the Internet). An intervehicular communication architecture was proposed in [3], whereas a prototype was built in [4]. The basic application of a VANET is to allow arbitrary vehicles to broadcast safety messages (e.g., road condition and traffic accident information) to nearby vehicles and RSUs such that other vehicles may adjust their traveling routes, and RSUs may inform the traffic control center to adjust traffic lights for avoiding possible traffic congestion. Some other applications allow a group of known vehicles (e.g., police cars and tour buses) to securely communicate among themselves.

Similar to the mobile phone network, after fulfilling basic functions, the trend is to make use of the framework to provide value-added services. One such service is querying service. One well-known example is the location-based querying service (e.g., seeking the location of the closest Chinese restaurant). We assume that one or more querying service providers (QSPs) are installed in the backbone. They have huge databases that contain information on points-of-interest (POIs) in its region or even in the whole city, and this information is collected using approaches similar to the technique in [5]. Drivers who pass by can issue queries about POIs to a QSP. Similar to existing location-based querying services for web applications [6], a querying service for VANETs may be a paid service. The charging model can be on a per-query basis to allow drivers who may not be a permanent resident in the city to use the service in an ad hoc manner. There should be a mechanism for recording which vehicles have used the querying service so that bills can be issued to them.

On the other hand, the user may not want anyone (including the infrastructure units such as RSUs), except for the QSP, to know the content of his/her query. To further protect a user’s privacy, although all RSUs, TAs, and QSPs collude, it should be impossible to link up a query with the real identity of the user. For example, a driver who issued a query to a QSP to ask for the location of the nearest nightclub may not want anyone to know about it. The same confidential and privacy-preserving issue occurs when making other kinds of queries. For example, a driver who issued queries to a QSP to get more information about a product may not want other users to know in which product he/she is interested. The difficulty of the problem is on how we can record or validate the user for using the service while making sure that the query and the real identity of the driver is not linkable. On the other hand, a user can use the service only once for each validation. This requirement makes the problem nontrivial. One simpler approach can be adopted if this requirement can be relaxed [7]. We will formally define our system model, assumptions, and security requirements in Section III.

Summary of Our Contributions: In this paper, we propose an OT-based Private Querying (OPQ) scheme for solving the confidential and privacy-preserving problem for querying services in VANETs. Our scheme is based on the techniques of pseudoidentity and indistinguishable credentials. To use the service, a driver first has to authenticate himself/herself to a nearby RSU. Then, the RSU passes his/her $N_s$ credentials, where $N_s$ is a system parameter, and we will evaluate the performance of our system as $N_s$ varies. Using the principle of oblivious transfer (OT) [8], the RSU cannot tell which $N_s$ credentials it has passed to the driver. By presenting the $N_s$ credentials
obtained to a QSP, a driver can accordingly issue its query. This way, although all RSUs, TAs, and QSPs collude, no one can link up a query with the real identity of the querier. Other basic security issues, e.g., message integrity and confidentiality, are also addressed in this paper. We will explain our scheme in detail in Section V. We show that our schemes are effective in terms of processing delay, message overhead, and success rate based on a simulation study.

II. RELATED WORK

The privacy-preserving querying problem for mobile devices (e.g., mobile phones) has been addressed in [9]–[11]. In [9], a simple public key infrastructure (PKI) is adopted, and an anonymity router (AR) is introduced between the mobile device and the QSP. In brief, the return address (e.g., mobile phone number) of the mobile device is encrypted using the AR’s public key, whereas the query is encrypted using the QSP’s public key. This solution seems to work, but it does not consider the case that the AR and the QSP collude. In VANETs, RSUs are located along the road; therefore, it is easier to be hacked.

To provide a higher level of privacy, we should have a solution that can still maintain the privacy, although the infrastructure units collude. The return addresses of mobile devices and their queries can easily be linked if the AR and the QSP share their information. In [10] and [11], it is assumed that, if the current location of the device (e.g., mobile phone) is known, the identity of the user will be revealed. Therefore, both parties try to hide the current location of the querying device. Rather than using a cryptographic approach, they use networking approaches. In [10], a mobile user who wants to use the service first collects the locations of some nearby devices to form a so-called cloaked spatial region. It then sends out the whole region to the QSP, which then returns the POI location that is closest to the cloaked spatial region. This solution suffers two drawbacks. First, the POI location is only the closest to the cloaked spatial region but may not be the closest to the querier. Second, in a VANET setting, it is not always possible to find other neighbors to form a cloaked spatial region. For example, the traffic density is usually very low on a countryside highway. In [11], a mobile user who wants to issue location-based queries first sends out a fake location. Upon receiving a reply from the QSP, it incrementally adjusts the fake location and sends out new queries. This process repeats until the result from the QSP is accurate enough for the user’s purpose. Obviously, such a repeated querying process requires very heavy transmission overhead and induces very heavy computation overhead at the QSP.

Providing a location-based querying service in a multihop mobile ad hoc network (MANET) environment has been addressed in a number of efforts, e.g., [12]. However, none of these efforts addresses the privacy preservation problem. Recently, in [13], a hierarchical approach has been proposed, and this approach can be extended to easily provide privacy preservation. The whole network is split into small regions, and each region has a head node, which maintains the information of all other nodes. Head nodes of different regions are then connected to higher level nodes. This process repeats, and a multilevel hierarchy is formed. In the querying process, each region in the hierarchy can then work as a cloaked spatial region, and thus, the privacy of the querier is preserved similar to the approach in [10]. Obviously this scheme is not suitable for a VANET environment, because in a VANET, vehicles move at high speed most of the time, and the network topology rapidly changes. Thus, it is impossible for a multilevel hierarchy to be maintained. A summary of all previous networking approaches that provide privacy-preserving location-based querying services can be found in a recent paper [14].

Our scheme is based on the idea of an indistinguishable (anonymous) credential. Such a credential system was introduced by Chaum [15]. The system allows a user to obtain a credential from one organization and later show the possession of the credential to another organization, whereas the transactions at the two organizations are not linkable. The idea of anonymous credential has been adopted in different applications. For example, [16] 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. The application that is closest to our approach is the anticollusion anonymous credentials scheme proposed in [17]. The selection of credential is also based on OT, but the requirements are different and, therefore, cannot easily be adopted to our case. First, the approach requires the transfer of a credential from one user to another to be inconvenient; therefore, each user has to pick a large number of credentials from an organization. The required complexity is very high for the highly dynamic VANET environment. On the other hand, nontransferability is not needed in our scheme. In our case, the credential that a user possesses should be used only once.

Other works related to the security issues in VANET include [18]–[23]. In [19], 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 IVC scheme was proposed. A vehicle relies on an RSU to verify the signature of another vehicle. In [20], group communications in VANETs are considered, and a group key update protocol was proposed. In [21] and [22], some security and privacy-enhancing communications schemes (SPECSs) 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. Most recently, in [23], an approach for privacy-aware traffic monitoring has been proposed based on the idea of data aggregation.

III. SYSTEM MODEL AND SECURITY REQUIREMENTS

A. System Model and Assumptions

Recall that we consider a vehicular network that consists of OBUs installed on vehicles and RSUs along the roads. A TA maintains the real identities of vehicles and a QSP, which is responsible for answering queries from vehicles. We further assume the following conditions.

1) The QSP is always online so that when a vehicle issues a query, the QSP can answer it in real time. Again, to
avoid being a single point of failure, redundant QSPs are installed. These QSPs periodically synchronize their databases.

2) RSUs, TAs, and QSPs communicate through a secure fixed network (e.g., the Internet).

3) There exists a conventional PKI for initial handshaking. Any RSU R periodically broadcasts its identity \( I_{DR} \) and conventional public key \( CPK_R \) with hello messages to vehicles that travel within the RSU-to-vehicle communication (RVC) range. Thus, \( CPK_R \) is known by all nearby vehicles.

4) The real identity of any vehicle is only known by the TA and itself but not by other parties.

5) A reasonably large number of queries are issued to the QSP. It is because, if there is only one query, the sender can easily be linked up with the query.

B. Security Requirements

We aim at designing a scheme for the provision of confidentiality and privacy-preserving querying service in VANETs. The following security requirements have to be satisfied.

1) Message authentication. A vehicle should be authenticated before it can issue a query.

2) Identity privacy preserving. The real identity of a vehicle should be kept anonymous from other vehicles, and a third party should not reveal a vehicle's real identity by analyzing multiple messages that it sent.

3) Traceability. Although a vehicle's real identity should not be hidden from other vehicles, the TA should obtain a vehicle's real identity so that it can be charged for using the querying service.

4) Confidentiality. The content of a query that is sent by a vehicle should be kept confidential. Only QSPs can read the contents.

5) Unlinkability. Although all RSUs, TAs, and QSPs collude, they cannot link a vehicle's query with its real identity.

IV. Preliminaries

A. Bilinear Maps

Our schemes are pairing based and are defined on two cyclic groups with a mapping called a bilinear map \([24]\). Let \( G \) be a cyclic additive group and \( G_T \) be a cyclic multiplicative group. Both groups \( G \) and \( G_T \) have the same prime order \( q \). The mapping \( \hat{e}: G \times G \rightarrow G_T \) is called a bilinear map if it satisfies the following properties.

1) Bilinear. \( \forall P, Q, R \in G \), and \( \forall a, b \in \mathbb{Z} \), \( \hat{e}(Q, P + R) = \hat{e}(P, Q) \cdot \hat{e}(R, Q) \). In addition, \( \hat{e}(aP, bP) = \hat{e}(P, P)^{ab} = \hat{e}(aP, P)^b = \hat{e}(P, P)^{ab} \).

2) Nondegenerate. There exists \( P, Q \in G \), such that \( \hat{e}(P, Q) \neq 1_{G_T} \).

3) Computable. There exists an efficient algorithm for computing \( \hat{e}(P, Q) \) for any \( P, Q \in G \).

The bilinear map \( \hat{e} \) can be constructed using pairings on elliptic curves. Each operation for computing \( \hat{e}(P, Q) \) is referred to as a pairing operation. The groups \( G \) and \( G_T \) are called bilinear groups. The security of our scheme 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 for obtaining \( a \), 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.

B. Oblivious Transfer (OT)

Our scheme also adopts the principle of OT when an RSU passes one or more credentials to a vehicle. The idea is explained as follows. An RSU passes \( N_c \) valid credentials to the vehicle, the vehicle is allowed to pick \( N_s \) of them only, whereas the RSU does not know which \( N_s \) credentials the vehicle has picked. OT has several variations, e.g., 1 out of 2, 1 out of \( n \), and \( m \) out of \( n \). In [8], there is a very recent work on OT. Interested readers can refer to it for technical details.

V. Our Solutions

This section presents our proposed scheme for providing a confidential and privacy-preserving querying service in VANETs.

Following [21] and [22], we assume that the parameters \( \{G, G_T, q, P, P_{pub}\} \) are generated and made public by the TA. In addition, each vehicle is assigned a real identity \( RID \) and a password \( PWD \) during network deployment or vehicle first registration. Furthermore, assume that each vehicle \( V_i \) has already performed an initial handshaking with a nearby RSU and TA and obtained the TA's master key \( s \), the shared secret \( m_i \) with the RSU, and the shared secret \( t_i \) with the TA. The RSU has also obtained \( V_i \)'s verification public key \( VPK_i = t_i \oplus RID \) from the TA for verifying \( V_i \)'s pseudoidentity in the future. Note that the TA assigns different \( t_i \) (and, thus, different \( VPK_i \)) when \( V_i \) enters different RSUs' ranges; therefore, although all RSUs collude, they cannot trace a vehicle's route.

Fig. 1 gives an overview of our scheme. There are the following five basic modules.

1) The QSP generates its public and private key pair.

2) The QSP generates credentials and passes a set of \( N_c \) credentials to each RSU.

3) The RSU verifies a vehicle's identity for billing purposes in the future.

4) The RSU passes the vehicle a subset of \( N_s \) credentials using the principle of OT.

5) The vehicle sends its query by presenting the subset of \( N_s \) credentials to the QSP. Then, the QSP answers accordingly.

Next, we will explain these modules one by one.

A. Key Generation by the QSP

A QSP generates a random number \( SK_q \) to be used as its private key. It then computes the corresponding public key as \( PK_q = (SK_q)P \). \( SK_q \) is kept private, whereas \( PK_q \) is publicly known by all RSUs and all vehicles.
the following two parts: 1) $ID_{11}$ and 2) $ID_{12}$. Here, $ID_{11} = rP_{pub}$ and $ID_{12} = VPK_i \oplus H(m_i.ID_{11})$. The corresponding signing key is $SK_i = (SK_{11}, SK_{12})$, where $SK_{11} = sm_i.ID_{11}$, and $SK_{12} = sH(ID_{12})$. $H(\cdot)$ is a MapToPoint hash function as aforementioned. It then generates the querying service request message $M_i = \{QSREQ\}$ and its signature $SIG_{SK_i}(M_i) = SK_{11} + h(M_i)SK_{12}$ on it. Here, $h(\cdot)$ is a one-way hash function such as secure hash algorithm 1 (SHA1) [26]. $V_i$ then sends $\langle ID_{11}, M_i, SIG_{SK_i}(M_i) \rangle$ to the RSU.

The RSU verifies signatures in querying service request messages in a batch (if it receives more than one querying service request message in a batch verification interval). With the pseudoidentity of each vehicle $V_i$, the RSU first extracts $V_i$’s verification public key $VPK_i$ and shared secret $m_i$ by checking which of the stored $(VPK_j, m_j)$ pairs (sent by TA previously) satisfies $ID_{12} = VPK_i \oplus H(m_i.ID_{11})$. Without loss of generality, assume the RSU receives the querying service request messages from $V_1, V_2, \ldots, V_n$ and it verifies their signatures by checking if $e(\sum_{i=1}^{n} SIG_{SK_i}(M_i), P) = e(\sum_{i=1}^{n} m_i.ID_{11} + h(M_i)H(ID_{12}), P_{pub})$. The RSU can also adopt the binary-search approach to check the validity of each individual signature.

If a vehicle’s signature is valid, a credential will be issued for it to use the QSP’s service. The RSU then records $V_i$’s pseudoidentity and the signing key used by $V_i$ on $h(V_i)$ and the signing key used by $V_i$ on $h(V_i)$ and the signing key used by $V_i$ on $h(V_i)$ and the signing key used by $V_i$ on $h(V_i)$ and the signing key used by $V_i$ on $h(V_i)$ and the signing key used by $V_i$ on $h(V_i)$. Checking which of the stored service request messages from $V_1, V_2, \ldots, V_n$ and it verifies their signatures by checking if $e(\sum_{i=1}^{n} SIG_{SK_i}(M_i), P) = e(\sum_{i=1}^{n} m_i.ID_{11} + h(M_i)H(ID_{12}), P_{pub})$. The RSU can also adopt the binary-search approach to check the validity of each individual signature.

If a vehicle’s signature is valid, a credential will be issued for it to use the QSP’s service. The RSU then records $V_i$’s pseudoidentity into a database. At a later time (e.g., every midnight), all pseudoidentities stored in the RSU’s database are transferred to the TA. The TA then computes vehicles’ real identities for issuing charging bills later on.

D. Credential Transfer by RSU

Assume that $V_i$’s signature is found to be valid. In addition, assume that the RSU possesses the set of credentials $C_{R0}, C_{R1}, \ldots, C_{RN_s-1}$. The RSU transfers $N_s$ credentials, where $N_s$ is a system parameter, to $V_i$ using the following steps.

1) The RSU picks a random number $SK_r$ to be used as its private key and computes $PK_r = (SK_r, h_0, h_1, \ldots, h_{N_s-1}, ESIG_{SK_r}(h_0), ESIG_{SK_r}(h_1), \ldots, ESIG_{SK_r}(h_{N_s-1}))$ and its signature $CSIG_{SK_r}(M_r)$. Here, $h_i$ is a one-way-hash [26] of the $i$th credential (i.e., $h_i = h(C_{R_i})$), and $ESIG_{SK_r}(h_i)$ is the QSP’s signature on $h_i$.

2) $V_i$ first verifies the RSU’s signature on $M_r$. Then, it randomly picks $N_s$ integers $a_0, \ldots, a_{N_s-1}$ in the range $[0, N_s - 1]$. It also picks another two random numbers $x$ and $k$, computes the points $xP$ and $kP$, and composes the message $M_i = \langle xP, Temp_{P1}, \ldots, Temp_{P1}(N_s-1) \rangle$ where $Temp_{P1} = kP + xP + a_iP$. It sends $\langle ID_{i}, M_i, SIG_{SK_i}(M_i) \rangle$ to the RSU. Note that the pseudoidentity and the signing key used by $V_i$ here are the same as in its $QSREQ$ message.

3) Upon receiving $xP$ and each $Temp_{P1}$ for all $i$ in the range $[0, N_s - 1]$ (recall that RSU does not know which $i$ vehicle $V_i$ has chosen), the RSU multiplies $xP$ by its secret key $SK_r$ and subtracts the product from
Temp_{ij} - iP. That is, it tries to obtain kP to be the decryption of Temp_{ij} - iP. It then includes Temp_{ij} = Cr_i + h(kP) into message M_r, which will be sent to V_i. After considering all V_i’s choices, M_r should contain N_s × N_sTemp_{ij} values. Finally, V_i sends to V_i \{ID_i, M_r, CSIG_{sk}(M_r)\}. Note that the RSU has to include ID_i in the message, because it can perform OT with more than one vehicle at the same time.

4) Upon ensuring that the RSU’s message is for it and verifying the RSU’s signature, because V_i knows the value of k’, which is equal to k that it has picked earlier, it can compute Cr_{a_i} = Temp_{ij} - h(kP) for all j in the range [0, N_s - 1]. Next, V_i verifies whether Cr_{a_i} = h(Cr_{a_i}) for all j in [0, N_s - 1] and validating the QSP’s signature on h_{a_0},...,h_{a_{N_s-1}}. The signature validation can be done by checking whether \(\hat{\epsilon}(ESIG_{sk}(h_{a_0})), P) = \hat{\epsilon}(H(h_{a_0}), PK_q)\) as \(\hat{\epsilon}(h(h_{a_0}), PK_q) = \hat{\epsilon}(H(h_{a_0}), PK_q)\).

5) To ensure that the RSU does not perform a same-message attack (once this attack has been done, an RSU can cheat V_i to pick certain credentials, as mentioned in [27]), it checks whether h_{a_i} \neq h(Cr_{a_i}) for all i \in [a_0,a_{N_s-1}] after using the QSP’s public key to verify ESIG_{sk}(h_{a_i}).

Note that, among the N_s - 1 deceptions that the RSU makes in step 3, only N_s of the k’ values (i.e., corresponding to the i chosen by V_i) matches the k value chosen by V_i, whereas all other values are garbage. Thus, V_i can only obtain the credentials Cr_{a_0},...,Cr_{a_{N_s-1}} but not other credentials.

Finally, the RSU stores V_i’s pseudoidentity (only one, although a number of messages from V_i are involved in the aforementioned process) and the corresponding shared secret m_i into its database. At a later time, all pseudoidentities and shared secrets stored in the RSU’s database are transferred to the TA. With V_i’s pseudoidentity ID_i and m_i, the TA can search through all the stored (RID_j,t_j) pairs from its repository. Vehicle V_i’s real identity is the RID_j value from the entry that satisfies the expression ID_{12} + t_j + H(m_i,ID_{13}) = RID_j as ID_{12} + t_j + H(m_i,ID_{13}) = t_j + RID_j + H(m_i,ID_{13}) + t_i + H(m_i,ID_{13}) = RID_j. No other party can obtain vehicle V_i’s real identity, because t_i is known only by the TA and V_i itself. Then, charging bills can be issued accordingly.

E. Query and Answer Forwarding

V_i then generates and stores a random number y that will be used as a temporary symmetric session key for the current querying session. The value of y is encrypted using the QSP’s public key PK_q = (SK_q,P), whereas the credential and its query QUERY are symmetrically encrypted using y. Finally, it sends the message \(\langle EENC_{PK_q}(y), ENC_y(Cr_i, QUERY)\rangle\) to the QSP through any RSU. Note that the value of y is only known by the QSP, and V_i’s query is encrypted using y. Thus, the QSP is the only party who knows what V_i is querying. In addition, note that, aside from the credential, V_i does not need to present any identity information (not even its pseudoidentity) in its query.

Upon receiving V_i’s query, the QSP first decrypts EENC_{PK_q}(y) using its secret key SK_q and obtains y. Then, it decrypts ENC_y(Cr_i,QUERY) using y and obtains the credential and the query. The QSP first checks from its database whether the same set of credential numbers presented by V_i have been presented by any other vehicle. If yes, it sends an error message \(\langle M_q, ESIG_{sk}(M_q)\rangle\), where M_q = \{Credential Reuse Error\}, to V_i. Upon getting this error, V_i will present its pseudoidentity used in its QSREQ message and the QSP’s signed error message to the RSU. After verifying the QSP’s signature on the error message, the RSU will repeat the steps in Section V-D to let V_i pick another N_s credential. Note that, by presenting the QSP’s signed error message, any RSU will add a remark to the database so that V_i will not be charged more than once for a single query. In Section VII, we will show that, if an RSU updates all its credentials on time, such a credential reuse error is quite unlikely.

If the QSP finds that the set of N_s credentials has not been presented by other parties, for each credential, it verifies whether \(Cr_{a_i} = \{Ra_i, ESIG_{sk}(Ra_i)\}\) is signed by it by checking whether \(\hat{\epsilon}(ESIG_{sk}(Ra_i), P) = \hat{\epsilon}(H(Ra_i), PK_q)\) as \(\hat{\epsilon}(h(Ra_i), PK_q) = \hat{\epsilon}(H(Ra_i), PK_q)\). After confirming that all N_s credentials presented by V_i are valid and the same set has not been presented by other parties, the QSP searches through its database and tries to answer V_i’s query. Because the objective of this paper is to ensure that V_i can securely and confidentially issue its query, we will not discuss how the QSP searches for answers associated with V_i’s query. See, e.g., [11]. Once the QSP has received the answer ANSWER to V_i’s query, it generates the reply M_q = \(\langle EENC_{PK_q}(y), ENC_y(ANSWER)\rangle\) and sends it to V_i through the RSU. Note that EENC_{PK_q}(y) is included in the reply and is used as the query’s pseudoidentity so that V_i knows that the QSP, indeed, answers its question. In addition, because all mirror QSPs have identical functionalities and databases, an RSU only needs to forward a query to the physically closest QSP to reduce the response time.

If, after a predefined interval, vehicle V_i still has not received the reply from the QSP, it assumes that either its query message for the QSP or the QSP’s answer for it has been corrupted by noise in the environment or blocked by obstacles, and it renders its previous query to the QSP.

VI. Security Analysis

We analyze our scheme with respect to the security requirements listed in Section III.

A. Message Integrity and Authentication

The signature SIG_{sk}(M_i) on message M_i by vehicle V_i is composed of SK_{i1} and SK_{i2}. SK_{i1} is defined as sm_{i1}ID_{i1}, where m_i is the shared secret between vehicle V_i and the RSU. Because m_i is securely transmitted from the RSU to V_i during initial handshaking, no one, except for V_i and the RSU, knows its value. Thus, no other vehicle knows how SK_{i1} can be composed. SK_{i2}, on the other hand, is defined as sH(ID_{i2}). Recall that ID_{i2} = VPK_i ⊕ H(m_iID_{i1}). Again, because no
other vehicle knows $m_i$ and, therefore, only $V_i$ can compute $SK_{i, 2}$, no other vehicle can forge a valid signature by vehicle $V_i$. Note also that RSUs do not know the master secret $s$ and thus cannot forge $V_i$’s signature.

B. Identity Privacy Preserving

Identity privacy preserving is an important feature of our scheme. We formally show that an attacker cannot easily obtain a vehicle’s real identity. Because the only information that is related to a vehicle’s real identity and is exposed in the network is its pseudoidentity, we show that an attacker cannot obtain a vehicle’s real identity, although it has the pseudoidentity. We argue that, if the decisional Diffie–Hellman (DDH) is hard, then the pseudoidentity of a vehicle can preserve its real identity. The proof is given as follows.

We first consider game 1 between a challenger and an attacker.

**Setup:** The challenger starts by giving the attacker a set of system parameters, including $P$ and $P_{pub}$.

**Choose:** The attacker then freely chooses two verification public keys $VPK_0$ and $VPK_1$ and sends them to the challenger (these choices do not need to be random, and the attacker can choose them in any way that it desires).

**Challenge:** The challenger sets a bit $x = 0$ with probability 1/2 and sets $x = 1$ with probability 1/2. The challenger then sends the attacker the pseudoidentity that corresponds to $VPK_x$ together with the group public key.

**Guess:** The attacker tries to guess the value of $x$ that is chosen by the challenger and outputs its guess $x'$.

The attacker’s advantage in this game is defined to be $Pr[x = x'] - 1/2$. We say that our pseudoidentity generation algorithm is semantically secure against a chosen plaintext attack (CPA) if the attacker’s advantage is negligible.

Next, we assume that we have an algorithm $A$ that runs in polynomial time and has a nonnegligible advantage $e$ as the attacker in game 1. We will construct game 2, in which a DDH attacker $B$ can make use of $A$ to achieve a nonnegligible advantage in breaking the DDH. $B$ is given a DDH instance $(P, aP, bP, T)$ as input and is asked to determine whether $T = abP$. We further let $t$ denote a bit that $B$ tries to guess (i.e., $t = 0$ for a positive answer $T = abP$, whereas $t = 1$ for a negative answer $T \neq abP$). Game 2 runs as follows.

**Setup:** Based on the DDH instance, $B$ makes up the parameters $(P, P_{pub} = aP)$ and gives them to $A$. Note that $a$ now plays the role of $s$ in our SPECS.

**Choose:** $A$ then chooses two verification public keys $VPK_0$ and $VPK_1$ and sends them to $B$.

**Challenge:** $B$ plays the role of challenger here; therefore, it randomly sets a bit $x$ and generates the pseudoidentity $ID = (ID_1, ID_2)$, where $ID_1 = raP$, $ID_2 = VPK_x \oplus H(rT)$, and $r$ is a random nonce and sends to $A$. (Note that $b$ now plays the role of the RSU vehicle shared secret $m_i$ in our SPECS.)

**Guess:** Finally, $A$ sends $B$ a bit $x'$ as its guess for $x$. $B$ positively answers the DDH problem that $T = abP$ if $B$’s guess is correct (i.e., $x = x'$).

Now, let us look at why $B$ can answer the DDH problem in this way. If $t = 0$ (i.e., $T = abP$), then $ID_2 = VPK_b \oplus H(rabP) = VPK_b \oplus H(bID_1)$ is a valid pseudoidentity in proper format. In this case, because $A$ has a nonnegligible advantage in the aforementioned game, it is likely that $A$ can break our SPECS system and correctly guess $x$ with probability $1/2 + \epsilon$. Thus, $Pr[B succeeds|t = 0] = 1/2 + \epsilon$. If $t = 1$, we claim that $Pr[B succeeds|t = 1] = 1/2$ only. To see why, we observe that, when $T$ is randomly chosen, the term $H(rT)$ in $ID_2$ cannot be canceled by the term $H(bID_1)$, and therefore, there is no way of obtaining $VPK_x$. Thus, the computation reveals no information about $x$. In this sense, the value of $x$ is hidden to $A$; therefore, although $A$ can break our SPECS system, the probability of correctly guessing $x$ is simply $1/2$ (by tossing a fair coin). Hence, $Pr[B succeeds] = 1/2 \times (1/2 + \epsilon) + 1/2 \times 1/2 = 1/2 + \epsilon/2$. Because $\epsilon$ is nonnegligible, $B$ can solve the DDH problem, but this case violates the assumption that DDH is hard. Therefore, our scheme is secure, because the pseudoidentity of a vehicle can preserve its real identity.

On the other hand, the random nonce $r$ makes the pseudoidentity of a vehicle different in different messages. This approach makes tracing the location of a particular vehicle over time difficult without the shared secret between the sender and the RSU. Furthermore, because the verification public key $VPK_i$ of a certain vehicle is different, as shown by different RSUs, although all RSUs collude, they have no way of tracing a particular vehicle’s traveling route.

C. Traceability and Revocability

Section V-C shows that the TA can trace a vehicle’s real identity; thus, traceability is satisfied.

D. Unlinkability

We first prove the correctness of the OT of credentials from RSU to a vehicle, because this case is the key to removing the linkage between a credential and its holder’s identity. Note that vehicle $V_i$ sends its choices to the RSU in an “encrypted” way in step 2. The message that it sends includes $Temp_{1,j} = kP + xPK_v + a_jP = (k + a_j + xSK_v)P$ for all $j$ in the range $[0, N_i - 1]$. Because the RSU does not know the values of $k$ and $x$, it has no way of obtaining the value of $a_j$. Therefore, the RSU does not know which credentials $V_i$ has obtained. Although it cooperates with the QSP, they have no way of linking up the credential used by $V_i$ and its identity.

E. Confidentiality

A randomly generated session key is encrypted using the QSP’s public key. The query and the answer are, in turn, encrypted using this session key. Thus, the QSP is the only party that knows what the query and the answer are.

VII. OBTAINING A NEW SET OF CREDENTIALS

In this section, we briefly discuss when an RSU should request for a new set of credentials from the QSP.
We illustrate the idea using a simple experiment as follows. We assume that the RSU handles querying service requests from 30 vehicles in a certain period. The RSU lets each vehicle pick \( N_c \) credentials from a pool of \( N_c \) credentials at random. We vary the value of \( N_c \) from 0 to 300 in steps of 10 (in real situations, the value of \( N_c \) should not be very large to limit the decryption time spent by RSU in OT) and try three values of \( N_c \) (1, 2, and 3) to study their impact on the average number of rounds that a vehicle needs to obtain \( N_c \) credentials such that the same subset have not been used. Recall that, if the QSP finds that the subset of \( N_c \) credentials presented by a vehicle has been used, it will ask the vehicle to go back to the RSU to get another \( N_c \) credential.

The result is shown in Fig. 2, in which each point of data is obtained from 100 random scenarios. When \( N_c \) is set to 1, 2, and 3, the minimum value of \( N_c \) required is 30, 6, and 5, and the corresponding numbers of rounds are 4.2, 4, and 1.4, respectively. A smaller \( N_c \) leads to an infinite round number. However, the round numbers stabilize to close to 1 when \( N_c \) is set to 90, 20, and 10 (i.e., 3, 2/3, and 1/3 times the number of vehicles) under the three cases. Thus, we suggest an RSU request for a new set of credentials after it served \( N_c \) credentials such that the same subset have not been used. Recall that, if the QSP finds that the subset of \( N_c \) credentials presented by a vehicle has been used, it will ask the vehicle to go back to the RSU to get another \( N_c \) credential.

VIII. SIMULATION RESULTS

In this section, we evaluate our scheme in terms of processing delay, message overhead, and success rate using a network simulation program. Our scheme is the first approach to address the privacy-querying problem in VANETs. Other related schemes (e.g., approaches that are designed for mobile devices and MANETs, as discussed in Section II) are under different settings and cannot be applied to VANETs. Therefore, our simulation aims at showing the performance of our scheme under different conditions. Through simulation, we show that the processing delay and message overhead caused by our cryptographic functions are acceptable. On the other hand, the success rate of our scheme is almost the same as transmitting a single normal message over wireless channel as the channel collision probability varies.

A. Simulation Models

In our simulation, we made use of two maps that are downloaded from the TIGER database [28]: One map is for Washington, DC, whereas the other map is for Texas. Washington, DC, represents a city road system in which most roads have a speed limit of 50 km/h. Texas represents a countryside road system in which some highways have speed limits of up to 120 km/h. RSUs are randomly placed onto each road. Considering speeding behavior, we assume that vehicles in Washington, DC, travel at speed that varies from 10 km/h to 70 km/h, whereas vehicles in Texas travel at speed that varies from 70 km/h to 140 km/h.

Some of the settings and parameters of our simulation are adopted from [18], [19], [21], and [22]. The RVC and IVC ranges are set to 600 and 300 m, respectively. In the backbone, there are TA and QSP servers. RSUs, the TA, and the QSP communicate with each other through a fixed infrastructure. The bandwidth of the DSRC channel and the fixed infrastructure are assumed to be 6 and 10 Mb/s, respectively. With regard to the processing time, following the benchmark in [29] and [19], we assume that each pairing operation takes 4.5 ms, each point multiplication over an elliptic curve takes 0.6 ms, each conventional asymmetric encryption takes 1.2 ms, whereas each conventional symmetric encryption takes only 0.6 ms. We further assume that the QSP requires a database checking time of 15 ms (roughly the time for three pairing operations) before it can answer a query.

Following [19], we set the sizes of the pseudoidentity, the elliptic-curve cryptography (ECC)-type signature, and the ECC-type public key to 42, 21, and 21 B, respectively. We further set the sizes of components that are newly introduced in our scheme and provide our reasons as follows. For a random number, we set its size to 21 B. With this length, two credentials may carry the same credential number only after about \( 4 \times 10^{50} \) credentials have been generated. This approach should be more than sufficient for VANET usage. Thus, the size of each credential becomes 42 B, because each credential is composed of a random number and an ECC-type signature. For control messages such as QSREQ and ACK, we set their size to 5 B. In our proposed scheme, the longest control message in symbolic format is of five characters only. In case a binary format is used, these 5 B can even support up to \( 2^{40} \) different control messages. Therefore, we argue that 5 B should be sufficient for control messaging in a VANET. For a query and an answer, we set their size to 255 B, which is equivalent to 255 characters. Our scheme is designed for short queries and answers, e.g., a short message service (SMS) in a mobile phone network [30]. Compared with the maximum size of 160 characters for an SMS, our 255 B can already support a much longer message.

The IEEE 802.11a standard is used to simulate the medium access control (MAC) layer. That is, when a vehicle wants to transmit, it first detects whether the channel is available. If the channel is continuously idle for a distributed coordination function interframe space (DIFS) duration, it starts its transmission. If the channel is found to be busy during the DIFS interval, it waits until the channel is available and defers its access for an extra period of \( \text{random}() \times aSlotTime \). Then, it senses the channel for another DIFS duration before it starts its transmission. The whole process repeats if the channel is again found to be busy. According to the IEEE 802.11a standard, DIFS is defined to be 34 \( \mu \)s, whereas aSlotTime is defined to be 9 \( \mu \)s. In our simulation, \( \text{random}() \) is a number in the range 0–9.
We define the collision probability of a wireless channel as the probability that the channel is occupied by other parties when a vehicle wants to transmit. A high channel collision probability value simulates the situation that several neighboring vehicles try to transmit at about the same time. For our simulation, we vary the collision probability from 0 to 0.32 in steps of 0.01 (we will show that, when the collision probability is greater than 0.32, even a single normal message transmission cannot be completed due to an infinite waiting time).

We perform the following two experiments: one experiment is for Washington, DC, and another experiment for Texas. In each experiment, we consider 1000 queries made by 1000 vehicles, whose speeds vary from 10 km/h to 70 km/h for Washington and from 70 km/h to 140 km/h for Texas. Each data point in Figs. 3–5 represents the average performance of these 1000 vehicles.

We define processing delay to be the duration from when a vehicle sends out its querying service request message to when it receives an answer from the QSP through a nearby RSU. In Fig. 3, we normalized the processing delay experienced by a vehicle by the duration that it stays in the range of the corresponding RSU (without loss of generality, we assume that a vehicle issues its querying service request message once it has entered an RSU’s range and that it is not blocked within the range concerned). For Washington, DC, a vehicle stays in an RSU’s range for 31–214 s, depending on its speed. Similarly, for Texas, a vehicle stays in an RSU’s range for 15–31 s. Message overhead, on the other hand, is defined as the actual data overhead induced by our scheme. One point that we must add here is that, to make each data point presentable, for both normalized delay and message overhead, we only consider data that is noninfinity. We define success rate as the percentage of noninfinity data among the 1000 vehicles (i.e., the percentage of queries that can be completed in reasonable time). In addition, we compare the cases with different values of $N_s$ (1, 2, and 3).

### B. Simulation Results

In the first set of experiments, we consider vehicles in Washington, DC. In Fig. 3, we can see that a longer delay is required if each vehicle is allowed to pick only one credential. This case can be explained using the following two reasons: 1) to reduce the hitting probability (the probability that two or more vehicles obtain the same credential), the credential pool at any RSU has to be large enough; however, this case implies that an RSU has to spend a longer time to encrypt more credentials for each OT session, and 2) referring to Section VII, although an RSU has 90 credentials, the average number of rounds required by a vehicle to obtain an unused credential is still more than the cases with $N_s = 2$ and $N_s = 3$. Thus, a longer time is needed for a vehicle to obtain an unused credential. On the contrary, if each vehicle is allowed to pick two or three credentials (the vehicle then presents all these two or three credentials to the QSP), the RSU only needs to keep 20 and 10 credentials in the pool, respectively, to achieve the same hitting probability. The significant save in encryption time here yields a better delay performance. Nevertheless, under all cases, the processing delay required by our scheme is less than 0.2% of the duration that the vehicle stays in an RSU’s range. Thus, there should be sufficient time to complete the whole querying process.

Fig. 4 shows the message overhead caused by our querying scheme in terms of bytes. The $N_s = 1$ case gives the highest message overhead and the $N_s = 2$ case gives the middle overhead, whereas the $N_s = 3$ case gives the lowest overhead. Such a difference is mainly caused by the number of credentials kept by each RSU. With more credentials in the pool, more hash values, more QSP signatures, and more encrypted values have to be transmitted from an RSU to the querying vehicle. Higher message overhead thus results.
Fig. 5 elaborates the success rate under different values of \( N_s \). Interestingly, it is shown that the success rate is almost the same under different values of \( N_s \). For comparison, we also include a line for the single normal (not caused by our scheme) message transmission case. It is easily shown that the impact of our scheme on the success rate is not significant, because, except at a channel collision probability of 0.33, the success rate of our scheme is the same as in single normal message transmission. That is, under the same channel collision probability, if a query cannot be completed with our scheme, a single normal message transmission also cannot be completed due to infinite waiting in the IEEE802.11a MAC scheme.

In the second set of experiments, we repeat all steps using the map of Texas. Recall that Texas mainly contains highways and that vehicles travel at much higher speeds that in Washington, DC. We also put the results in Figs. 3–5 for easier comparison. We can see that, in Texas, the delay performance has the same trend as in Washington, DC: The fewer the number of credentials that a vehicle is allowed to pick, the longer the delay that is required due to the aforementioned two reasons. Although a vehicle leaves an RSU sooner than vehicles in Washington, DC, the processing delay required is still less than 0.4% of the duration that the vehicle stays in an RSU’s range. Thus, there should be sufficient time for a vehicle to complete the whole querying process.

The message overhead performance and the success rate performance (as shown in Figs. 4 and 5, respectively) are almost the same as in the Washington, DC case, and therefore, we skip the explanations here.

In short, by allowing a querying vehicle to obtain more than one credential from an RSU (the vehicle then presents all these vehicles to the QSP), an obvious improvement in terms of processing delay and message overhead can be obtained. Nevertheless, there are two minor drawbacks as follows: 1) the querying vehicle needs to perform more processing on the credentials (more decryptions and verifications), but because the decryptions and verifications of credentials are performed by a computer device but not by tamper-proof device, there should be no affordability problem, and 2) the QSP needs to keep more information and, possibly, more complicated processing (instead of one credential, it needs to store and process a subset of credentials for each querying vehicle), but because the QSP is a server in the backbone, processing power, memory, and storage should not be critical issues.

IX. Conclusion

We have proposed a scheme for providing a confidential and privacy-preserving querying service in VANETs. A vehicle can issue any kind of query without leaking its real identity to anyone using our scheme. A vehicle only needs to present a subset of RSU-indistinguishable credentials to the QSP to use the service. With the principle of OT, no one can link up a vehicle’s query and its real identity, although all RSUs, TAs, and QSPs collude. Based on a simulation study, we show that our scheme is effective in processing delay, message overhead, and success rate. For future work, we will implement our scheme on a testbed. In addition, we will consider other secure applications in VANETs, such as secure navigations and on-VANET shops.

REFERENCES


Lucas C. K. Hui (SM’03) received the B.Sc. and M.Phil. degrees in computer science from the University of Hong Kong, Pokfulam, Hong Kong, and the M.Sc. and Ph.D. degrees in computer science from the University of California, Davis.

He is the Founder and an Honorary Director of the Center for Information Security and Cryptography, University of Hong Kong, where he is concurrently an Associate Professor with the Department of Computer Science. His research interests include information security, computer crime, cryptographic systems, and electronic commerce security.

Dr. Hui is a member of the Hong Kong Institution of Engineers.

Victor O. K. Li (F’92) received the SB, SM, EE, and Sc.D. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 1977, 1979, 1980, and 1981, respectively.

In February 1981, he joined the University of Southern California (USC), Los Angeles, where he became a Professor of electrical engineering and the Director of the USC Communication Sciences Institute. Since September 1997, he has been with the University of Hong Kong, Pokfulam, Hong Kong, where he is currently the Chair Professor of information engineering with the Department of Electrical and Electronics Engineering. He has also served as the Managing Director of Versitech Ltd. He has been appointed to the Hong Kong Information Infrastructure Advisory Committee by the Chief Executive of the Hong Kong Special Administrative Region. He is a part-time member of the Central Policy Unit of the Hong Kong Government. He also serves on the Innovation and Technology Fund (Electronics) Vetting Committee, the Small Entrepreneur Research Assistance Program Committee, the Engineering Panel of the Research Grants Council, and the Task Force for the Hong Kong Academic and Research Network Development Fund, University Grants Committee. He was a Distinguished Lecturer with the University of California at San Diego; the National Science Council of Taiwan; and the California Polytechnic Institute, San Luis Obispo. He is a Codirector of the Area of Excellence in Information Technology funded by the Hong Kong government. Sought by government, industry, and academic organizations, he has extensively lectured and consulted worldwide. He was an Editor of Telecommunication Systems. He was a Guest Editor for Computer Networks and ISDN Systems. He is currently an Editor of ACM/Springer Wireless Networks. His research interests include information technology, particularly all-optical networks, wireless networks, and Internet technologies and applications.

Prof. Li is a Fellow of the Hong Kong Institution of Engineers and the International Association of Engineers. He was the Chair of the Computer Communications Technical Committee, IEEE Communications Society, from 1987 to 1989 and of the Los Angeles Chapter of the IEEE Information Theory Group from 1983 to 1985. He is a Cofounder of the International Conference on Computer Communications and Networks and was the Chair of its Steering Committee from 1992 to 1997. He was also the Chair of various international workshops and conferences, including, most recently, the 23rd IEEE Conference on Computer Communications in 2004 and the 2005 IEEE Workshop on High-Performance Switching and Routing. He has served as an Editor of IEEE NETWORK and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS (JSAC) Wireless Communications series. He was a Guest Editor for special issues of IEEE JSAC and the KICS/IEEE JOURNAL ON COMMUNICATIONS AND NETWORKING. He is currently an Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He has also delivered keynote speeches at several international conferences and the technology transfer and commercial arm of the University of Hong Kong (http://www.versitech.com.hk/) and on various corporate boards. He has received numerous awards, including, most recently, the People’s Republic of China Ministry of Education Changjiang Chair Professorship from Tsinghua University, Beijing, China; the U.K. Royal Academy of Engineering Senior Visiting Fellowship in Communications; the Outstanding Researcher Award from the University of Hong Kong; the Croucher Foundation Senior Research Fellowship; and the Order of the Bronze Bauhinia Star from the Government of Hong Kong.