No One Can Track You: Randomized Authentication in Vehicular Ad-hoc Networks

Wei Jiang
Department of Computer Science
Missouri University of Science and Technology
Email:wjiang@mst.edu

Dan Lin
Department of Computer Science
Missouri University of Science and Technology
Email: lindan@mst.edu

Feng Li
Department of Computer Science
Missouri University of Science & Technology
Email: lftrd@mst.edu

Elisa Bertino
Department of Computer Science
Purdue University
Email: bertino@cs.purdue.edu

***Abstract*—Vehicular Ad-hoc Networks (VANETs) are formed by a huge number of vehicles which act as the network nodes and communicate with one another. This emerging paradigm has opened up new business opportunities and enables numerous applications ranging from road safety enhancement to mobile entertainment. A fundamental issue that impacts the successful deployment of VANET applications is the security and privacy concerns raised by VANET users. However, it is a challenging task to authenticate vehicles while fully preserving their privacy. In this work, we propose a novel privacy-preserving randomized authentication protocol that leverages Homomorphic encryption to allow each individual vehicle to self-generate any number of authenticated identities to achieve full anonymity in VANETs. The proposed protocol prevents vehicles from being tracked by any single party including peer vehicles, service providers, authentication servers, and other infrastructure. Meanwhile, our protocol also provides traceability in case of any dispute. We have conducted both security analysis and experimental study which demonstrates the superiority of our protocol compared to other existing works.**

I. Introduction

In Vehicular Ad-hoc NETworks (VANETs), vehicles act as the network nodes, and communicate with one another to share information. Considering the large number of vehicles on roads, a variety of new services are envisioned, ranging from driving safety enhancement [1], dynamic route planning [2], to mobile entertainment [3]. For example, a vehicle may send inquiries to vehicles around certain landmarks to obtain the up-to-date traffic situation, the condition of a road, or parking information; passengers in vehicles can exchange files or chat with people in other vehicles along the trip via VANETs.

One of the key component towards the successful roll-out of VANET applications is to provide security and privacy guarantees. Otherwise, the rich functionality and services provided by VANETs may be abused, jeopardizing the safety of drivers and passengers. For example, a malicious vehicle can claim a fake traffic jam to gain the right of the road and cause other vehicles to make an unnecessary detour. Therefore, vehicles should be authenticated before they are allowed to exchange messages in VANETs.

Meanwhile, users' privacy should be preserved during authentication. Specifically, their real identities should be kept private and their locations should not be disclosed to the servers [4]. Otherwise, the authentication server may obtain the behavior pattern or track the user locations by keeping the records when and where the user requests for authentication. Similarly, peer vehicles may also be able to track each other by linking users with the same pseudonyms. On one hand, such server-wise and peer-wise privacy concerns should all be addressed in VANET applications. On the other hand, VANET application should still ensure traceability whereby law enforcement authorities are able to reveal the locations that the suspect vehicle has been to when disputes occur. Privacy preservation and traceability are two seemly conflicting requirements and hence it is one of the critical challenges that we aim to address in this work.

At a first glance, one may feel that the aforementioned security and privacy concerns resemble those encountered in other communication networks, especially Mobile ad-Hoc Networks (MANETs). However, compared to MANETs, VANETs have a larger number of nodes with high mobility and hence we are dealing with a history of locations; the nodes in VANETs have sufficient computational capability attributed to the on-board computers, and do not have power limitation as that in MANETs. Due to these differences in the network environment, solutions proposed in MANETs or other types of networks may not be suitable for VANETs.

In this paper, we present a novel authentication protocol called Randomized AUthentication (RAU) that truly preserves vehicles' privacy while still ensure traceability. The proposed protocol is designed based on Homomorphic encryption [5]. Using RAU, vehicles will be able to easily generate a new identity for each newly established communication. These randomized identities can be verified through the collaboration of a pair of authentication servers while each authentication server would not know the real identity of the authentication

978-1-5090-4327-9/17/$31.00 ©2017 IEEE

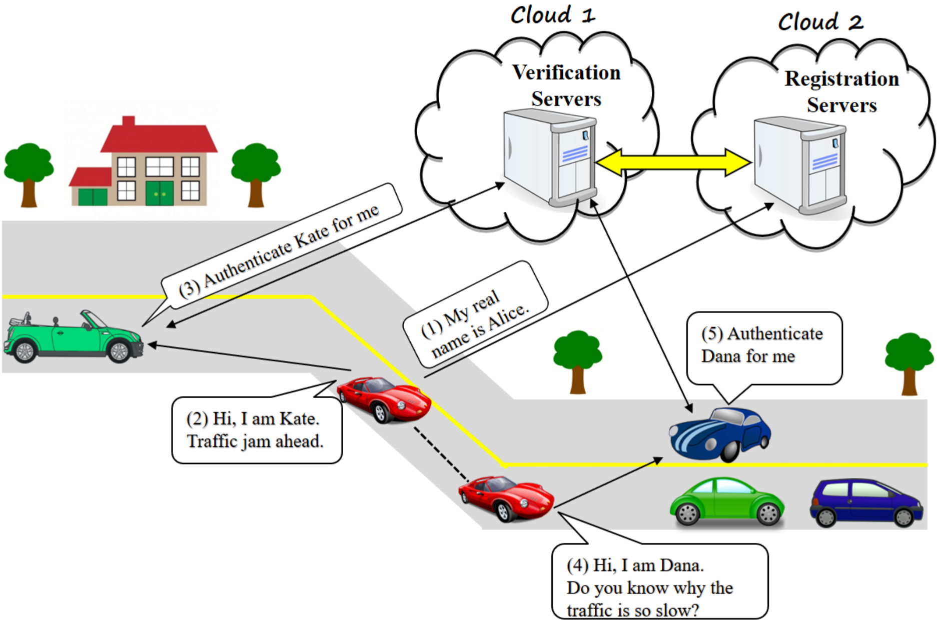
Cloud 1
Verification Servers
Cloud 2
Registration Servers
(3) Authenticate Kate for me
(1) My real name is Alice.
(5) Authenticate Dana for me
(2) Hi, I am Kate. Traffic jam ahead.
(4) Hi, I am Dana. Do you know why the traffic is so slow?

Fig. 1. A Scenario of the Proposed Randomized Authentication in VANETs. (1) The red sports car first registered her real identity 'Alice' at one authentication server. (2) The red car self-generates a new ID 'Kate' to communicate with the green convertible. (3) The green car asks another authentication server to verify if 'Kate' is a legitimate VANET user. Here the authentication server alone will only know that 'Kate' is a legitimate user but does not know 'Kate' is 'Alice' and hence cannot track 'Alice'. (4) The red car then uses another new ID 'Dana' to communicate with a blue car. (5) The blue car requests the authentication server to verify 'Dana'. Again, the authentication server would not know 'Dana' is actually 'Alice'.

requester. Figure 2 shows a simple example scenario. For traceability, the pair of authentication servers will execute a collaborative protocol so that the real identity of the malicious vehicle can be identified. In this way, not any single party in VANETs is able to track the VANET user.

Compared to existing VANET authentication works [6]–[10], our proposed RAU has a number of advantages. First, the RAU does not require any pre-generation of a long list of pseudonyms which could cause complicated ID revocation problem. Second, the RAU does not need the server to generate pseudonyms every time which prevents pseudonym generators, such as road-side units or group managers (i.e., peer vehicles) used in other works, from tracking the vehicles. Third, the RAU does not require the availability of road-side units which are not widely available in the real world due to deployment cost. Fourth, the RAU is very efficient which meets the real-time constraints in VANET applications very well. More detailed security analysis and performance studies will be presented in the remaining of the paper.

The rest of the paper is organized as follows. Section II reviews related works and Section III introduces preliminaries. Section IV presents the proposed privacy-preserving authentication protocol. Section V conducts security analysis. Section VI reports the experimental results. Finally, Section VII concludes the paper.

II. Related Work

Existing works on privacy-preserving authentication in VANETs can be classified into two main categories: (i) pseudonym-based protocols; and (ii) group-based protocols.

The general goal of the pseudonym-based authentication protocols is to enable vehicles to use different pseudonyms during communication rather than using their real identities. One of the earliest work in this category is by Raya and Hubaux [9]. They suggested that when a vehicle needs to sign a message, it randomly selects a private key from a huge pool of certificates issued by the authority. The message receiver will verify the sender's signature by checking the validity of the corresponding public key certificate. The problem of this protocol is that vehicles need to check a long list of revoked certificates when verifying each received signed-message, which is very time consuming. Raya, et al. in [11] proposed efficient revocation schemes. However, these schemes violate the location privacy requirement and are subject to a movement tracking attack [12]. In order to reduce the average overhead of message authentication, Calandriello et al. [13] proposed a hybrid scheme, which is also computationally expensive because it needs to check if the group signature is from a revoked vehicle [14]. Other pseudonym-based protocols can be found in [8], [10], [15]–[19], achieving different degrees of improvement over the key revocation problem. However, in most these protocols, the identity management authority is required to maintain the certificates associated with each vehicle so as to retrieve the vehicles' real identities when disputes occur. This allows the authority to track the vehicles' movement; hence, the vehicles' privacy is not fully preserved.

Another category of privacy preserving authentication proto-

978-1-5090-4327-9/17/$31.00 ©2017 IEEE

cols is group-based [6], [20]–[22]. The typical idea is to utilize group managers to group and authenticate vehicles, which enables vehicles to anonymously communicate with group members. Many group-based protocols leverage the group signature scheme [23]. Under the group signature scheme, vehicles can only verify that the messages are from a valid group member but do not know who is the actual sender, and hence vehicles are anonymous to their group members. For example, in the ECPP protocol proposed by Lu et al. [6], RSUs serve as the group manager who assigns the group keys to passing vehicles. The security and privacy of ECPP are later strengthened by Jung et al. [21] whose protocol guarantees unlinkability and traceability when multiple RSUs are compromised. Since the computation cost of group signature scheme is very high, some techniques have been proposed to improve efficiency, such as the distributed key management framework by Hao et al. [24] and the decentralized certificate authority with the biological-password-based two-factor authentication by Wang et al. [25]. Besides group-based signature scheme, other techniques have also been proposed to achieve anonymity within a group. For example, Zhang et al. [8] adopted the k-anonymity concept for preserving user privacy so that a vehicle is indistinguishable from $k-1$ other vehicles. However, k-anonymity requires at least k vehicles in vicinity which may not always be feasible in areas with few vehicles. In [7], Squicciarini et al. proposed a PAIM protocol which dynamically constructs groups via pure vehicle-to-vehicle communication, and leverages Pedersen commitment and secret sharing scheme to achieve anonymously authentication of vehicles. However, the proposed protocol requires complicated group management strategy which introduces extra overhead to the system. In addition, there have been some works [26]–[29] developed based on the ring signature or blind signature [30] for privacy-preserving authentication.

In general, the existing group-based protocols have at least one of the following disadvantages. First, the group manager has all the knowledge about group members and hence is able to track them. Second, the process of group updates and membership revocation is usually very costly due to the large number of vehicles and high mobility of vehicles. Third, the communication is constrained within group members. This requires an efficient and dynamic grouping algorithm which currently is still a challenging issue. Moreover, those protocols relying on the presence of infrastructure support (e.g., RSUs) may not be feasible in reality where RSUs rarely exist. Most recently, some hybrid approaches like CACPPA [31] have been proposed, which utilizes both the concept of pseudonym-based approaches and group-signature based approaches. They also use a cloud authority but it is different from our work whereby we utilize multiple cloud authorities to achieve separation of duty.

Finally, we would like to review the anonymous credential (AC) system [32]–[35] which has been chosen as a comparison approach in our experiments due to its popularity and similar setting to our problem. The AC system carries the same goal of providing non-transferable anonymous authentication to guarantee user anonymity. However, the AC system has the following limitations that have been overcome by our work. First, the AC system uses zero-knowledge proofs to verify if a user possesses a valid credential. Zero-knowledge proofs are computational expensive which leads to an inefficient verification protocol of the AC system. Second, the AC system is susceptible to both the man-in-the-middle and replay attacks whereas our protocol is robust against these attacks. Third, a valid credential in the AC system may be shared among malicious users who are not authorized to use the credential to obtain services. To solve the credential sharing problem, hardware based solutions have been proposed in [36]–[38]. In our proposed approach, we do not assume a user's computing device is equipped with such specialized hardware. Further, the AC system does not have an efficient and effective method to revoke a credential. The credential revocation problem was discussed in [39], but the solution only works when the system is adopted as a regular credential system (without randomizing a user's credential for each authentication). There still does not exist a concrete solution to the credential sharing problem in the actual anonymous credential system. There are other extensions [34], [35], but none of them directly addresses the aforementioned disadvantages.

III. Preliminary

For a better understanding, we first briefly review the additive homomorphic probabilistic public key encryption (HEnc$^+$) system which is the building block of the proposed authentication system.

Let E_{pk} and D_{sk} be the encryption and decryption functions in an HEnc$^+$ system with public key pk and secret key sk. Without sk, no one can discover x from $E_{pk}(x)$ in polynomial time. When the context is clear, we will omit pk and sk from the notations of the encryption and decryption functions. The HEnc$^+$ system has the following properties:

- The encryption function is additive homomorphic in that the product of the encryptions of x_1 and x_2 produces the encryption of $x_1 + x_2$.

$$E(x_1) * E(x_2) = E(x_1 + x_2) \quad (1)$$

- Given a constant c and $E(x)$:

$$E(x)^c = E(c * x) \quad (2)$$

- The encryption function has semantic security as defined in [40], i.e., a set of ciphertexts do not provide additional information about the plain-text to an adversary. E.g., suppose that y_1 and y_2 are the ciphertexts generated by performing the encryptions of x at different times using the same key, there is very high probability that $y_1 \neq y_2$, but $D(y_1) = D(y_2)$ holds.

Any HEnc$^+$ system is applicable, but in this paper, we adopt Paillier's public-key homomorphic encryption system [5] for the actual implementation due to its efficiency. In Paillier, the public key is $N = p * q$, where p and q are large primes with similar size, and they are private information. In general,

978-1-5090-4327-9/17/$31.00 ©2017 IEEE

the size of N should be at least 1,024 bits. The encryption function is defined as follows for x:

$$E(x,r) = (N+1)^x * r^N \mod N^2$$

where r is randomly chosen from $\mathbb{Z}^*_{N^2}$. Note that the encryption function is only based on the public key, and the group $\mathbb{Z}^*_{N^2}$ contains the elements from $\mathbb{Z}_{N^2} = \{0, 1, 2, \ldots, N^2-1\}$ which are co-prime to N^2. Since r is randomly selected each time a value is encrypted, $E(x, r_1) \neq E(x, r_2)$ if $r_1 \neq r_2$. On the other hand, $D(E(x, r_1)) = D(E(x, r_2)) = x$ regardless the value of r_1 and r_2.

IV. RAU: Randomized Authentication System

In this section, we first present the system overview and then the thread model, followed by the details of the protocols.

A. An Overview of the System

The RAU system consists of two major types of entities: users and authentication servers. Users are passengers in the car who would like to communicate with others via VANETs. There are two authentication servers residing in two different clouds, which are Registration Server (RS), and Verification Server (VS). The two servers collaborate with each other to conduct privacy-preserving user authentications, and hence none of them would be able to track the user alone. We assume users can communicate with the servers via Internet.

When designing each specific protocol, we aim to achieve the following security requirements of the anonymous authentication system:

- Prevent users from being tracked: This includes two aspects. First, the real identity of a legitimate user should not be known by other peer users in VANEts. Other peer users and any single authentication server would not be able to track the users' movement (i.e., a series of locations that the user has been to) by linking multiple authentication messages to the same user.
- Providing traceability: If necessary and under lawful request, the two authentication servers will be able to collaboratively reveal the real identity of a malicious user.

The proposed authentication system has three main phases: (1) user registration, (2) user authentication, and (3) identity tracing. Figure 2 illustrates an overview of the data flow in the system. At the beginning, users register at the RS server. The RS server shares an initial randomized authentication ID of each user with the VS server. Whenever users want to communicate with others, they can randomly generate pseudo identities which can be verified by the VS server. If there is any dispute, the two servers will conduct a tracing protocol to figure out the real identity of a malicious user. The detailed steps in each phase will be presented in the following subsections.

B. Threat Model

In our system, we adopt the following threat or adversary model.

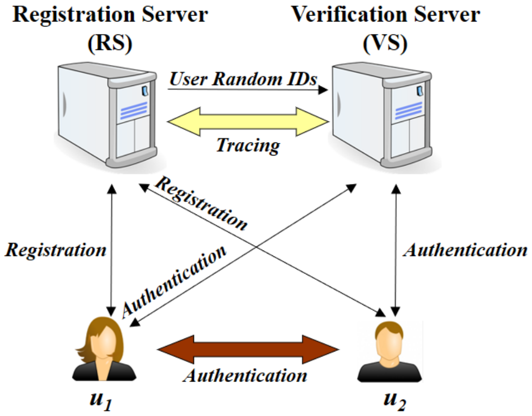

Fig. 2. An Overview of the Data Flow in RAU

- Like all existing work discussed in Section II, we assume that the two authentication servers adopted in the our authentication system are semi-honest. That is they follow the prescribed procedures of the proposed protocols and do not collude. This is a legitimate assumption if the two servers reside in two different well-known cloud platforms such as Amazon EC2 and Microsoft Azure which have not financial incentive to collude to damage their reputations.
- The users and service providers can be malicious. A malicious user can impersonate another user. A service provider may also be considered as a user who intends to obtain the needed services from other service providers.
- When the users are malicious, we will consider three common attacks under most authentication systems: man-in-the-middle, replay and credential sharing.

According to the above assumption, we will prove the security guarantee of the proposed authentication system according to the formal security definition of computational indistinguishability [41].

Definition 1 (Computational indistinguishability): Let k be a security parameter, and $X = \{X(a,k)\}_{k \in N \wedge a \in \{0,1\}^*}$ and $Y = \{Y(a,k)\}_{k \in N \wedge a \in \{0,1\}^*}$ be two distribution ensembles. Then X and Y are *computationally indistinguishable*, denoted by $X \overset{c}{\equiv} Y$, if for every probabilistic polynomially bounded adversary A, there exists a negligible function $\mu(\cdot)$ such that for every $a \in \{0,1\}^*$

$$|\text{Prob}[A(X(a,k)) = 1] - \text{Prob}[A(Y(a,k)) = 1]| < \mu(k)$$

C. User Registration

The registration phase is for a vehicle (user) to be authenticated by the server and obtain an initial random ID, based on which the user would be able to self-generate any number of random IDs to communicate with other peer vehicles.

It is worth noting that all communication in VANETs is via secure channels. Specifically, as a one-time setup, the registration server (RS) generates its own public-private key pair using the Paillier encryption scheme [5], and the public

978-1-5090-4327-9/17/$31.00 ©2017 IEEE

key is known by all entities in the VANETs. A session key between a user and a server, or between the two servers, can be generated using any well-known method, e.g., Diffie-Hellman key exchange protocol [42], to build the secure communication channel.

A new user can join the VANET system at any moment. To register, a user u sends certain identification information (ID_u) such as driver license number [1] to the registration server (RS) via the secure channel. If needed, the RS server can further verify u's identification information via a third party (e.g., an agency who performs background check for credit card applications). How to achieve robust identify verification is out of the scope of this paper, but the RS server can use any existing solutions.

The RS server computes an initial randomized authentication ID (RID_u^0) for user u as follows:

$$\mathrm{RID}_u^0 = E(\mathrm{ID}_u, r_u^0) \tag{3}$$

where $E(\mathrm{ID}_u, r_u^0)$ is a Paillier encryption of the identity of u with a random number r_u^0 using the RS' public key. RID_u^0 is sent to both user u and the verification server (VS). Since RID_u^0 is encrypted using the RS server's public key, only the RS server is able to decrypt it and reveal the real identity of the user. The actual identity of the user is always kept secret from the verification server during the lifetime of the user.

After user u is registered, both the RS and VS servers store the user's initial randomized authentication ID $\mathrm{RID}_{\mathrm{u}}^0$ in their local databases respectively. The plain texts of the real identities are discarded by the RS server to prevent attackers from hacking the system and stealing the sensitive information. The registration protocol is illustrated in Figure 3.

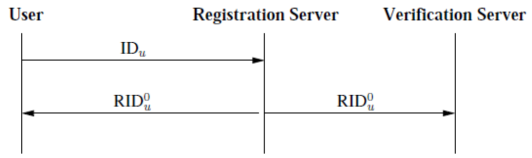

Fig. 3. User Registration (Please note that we show only the message content here. All messages are in fact encrypted.)

D. User Authentication

Without loss of generality, we present the one-way authentication protocol for user u_2 to verify if u_1 is a legitimate user. To achieve mutual authentication, the process can be executed again with u_1 and u_2 by switching their roles. The authentication protocol consists of three phases: (i) identity validation, (ii) ownership validation, and (iii) generation of a new randomized authentication ID. It is worth noting that if needed, u_1 can also perform concurrent authentication sessions with multiple users using this protocol.

*1) **Identity Validation**:* After the registration, the user u_1 obtained one initial randomized ID from the authentication server. User u_1 can use it directly for the follow-up communication with other vehicles if u_1 is no longer at the same location where it obtained the randomized ID, or u_1 can generate another new randomized ID based on this initial ID using the protocol in Section IV-D3.

Let $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u}_1}$ denote the randomized ID that u_1 will use to authenticate himself with user u_2. The following steps will be performed (illustrated in Figure 4):

- **Generating a shared random number**
  u_1 executes the Diffie-Hellman key exchange protocol with u_2 to mutually generate a shared random number $k_{u_1 u_2}$. The protocol guarantees that the probability of other two users obtaining the same random number $k_{u_1 u_2}$ is close to zero as long as one of the users follows the protocol. In other words, $k_{u_1 u_2}$ is unique for each pair of users each time they execute the protocol. The use of this random number is to prevent the man-in-the-middle attack (discussed in Section V).
- **Setting up a pending request by u_1**
  Before sending $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u}_1}$ to user u_2, u_1 will first register a pending authentication request at the VS server by sending the message: $p_{u_1} = [\mathrm{RID}^{\mathrm{i}}_{\mathrm{u}_1}, k_{u_1 u_2}]$. The VS server will search its database to look for $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u}_1}$. If $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u}_1}$ exists, then the VS server will check if there is a duplication of this $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u}_1}$ in the pending request table (to avoid man-in-the-middle attack, described in Section V). If there is no such randomized ID in the pending request table, the VS server will record this pending request. On the other hand, if it does not exist in the database or there is a duplication in the pending request table, the VS server will deny the authentication request.
- **Exchanging the randomized ID**
  Upon receiving the acknowledgment of successful registration of the authentication request from the VS server, u_1 sends $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u}_1}$ to u_2.
- **Verifying the randomized ID**
  For u_2 to verify the received $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u}_1}$, user u_2 forwards this randomized ID together with the random number $k_{u_1 u_2}$ to the VS server. If the VS server finds a pending authentication request that matches the message sent by u_2, the VS server will inform u_2 that this is a valid ID. Otherwise, the VS server will inform u_2 that authentication fails.

*2) **Ownership Validation**:* Once user u_2 confirms the validity of $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u}_1}$, user u_2 may need to further verify whether user u_1 is the real owner of the randomized ID. This step prevents an adversary from using the stolen $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u}_1}$ in order to prevent common attacks such as man-in-the-middle, replay and credential sharing (analyzed in Section V). For instance, an attacker may hack into the server's system or the user's device to obtain a copy of the current randomized IDs of some users, not the private key which is assumed to be securely stored. Unless the malicious user continuously monitors or

[1] Here we use driver license number for illustration only. Other information that verifies a user's identity such as SSN can also be used.

978-1-5090-4327-9/17/$31.00 ©2017 IEEE

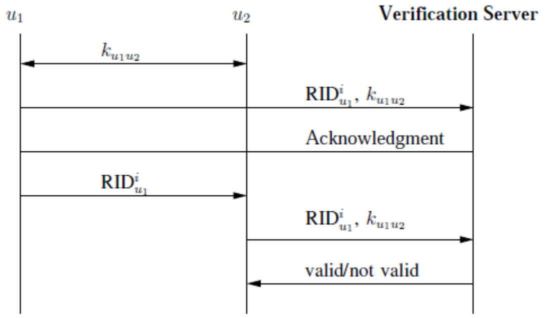

Fig. 4. Identity Validation

fully controls the server's system which is usually very difficult not to be detected by the server, he/she would not be able to impersonate other users using the obtained one-time randomized IDs because he/she cannot successfully pass the following ownership validation step. (Note that this step is optional and not necessary when the user and the servers' systems are reasonably secure.)

- **Generating a random challenge**
  User u_2 selects two random values c and r, and sends the following value v_1 to u_1.
  $$v_1 = \left(\text{RID}^{\text{i}}_{\text{u}_1}\right)^c * E(0, r) \tag{4}$$
  where c is a challenging value for u_1 to discover, and r can be any random number just for performing the encryption of 0. The purpose of multiplying with $E(0, r)$ is to randomize $\left(\text{RID}^{\text{i}}_{\text{u}_1}\right)^c$, so that it is computationally infeasible for an adversary to compute the discrete log of $\left(\text{RID}^{\text{i}}_{\text{u}_1}\right)^c$ to obtain c (Section V-A provides detailed security analysis on this regard).
- **Solving the random challenge**
  Only if user u_1 is the real identity owner, u_1 will be able to compute the encrypted value of the challenging value c. Specifically, u_1 first encrypts the multiplicative inverse of his or her real identity $\text{ID}^{-1}_{u_1}$. Then, u_1 computes a value v_2 by $v_1^{\text{ID}^{-1}_{u_1}}$. According to the properties of homomorphic encryption, value v_2 is equal to the encrypted value of c as deduced as follows:
  $$\begin{aligned} v_2 &= v_1^{\text{ID}^{-1}_{u_1}} \\ &= \left(\left(\text{RID}^{\text{i}}_{\text{u}_1}\right)^c * E(0, r)\right)^{\text{ID}^{-1}_{u_1}} \\ &= E(c * \text{ID}_{u_1} * \text{ID}^{-1}_{u_1}, r') \\ &= E(c, r') \end{aligned}$$
- **Verifying Ownership**
  Then, u_1 sends v_2 along with k_{u_1,u_2} to VS who will ask the RS to decrypt $v_2 = E(c, r')$ and obtain an decrypted value $D(E(c, r'))$. In addition, u_2 needs to send c and k_{u_1,u_2} to VS in parallel. VS will forward the v_2 and c with the same k_{u_1,u_2} to RS. Then, RS will decrypt $v_2 = E(c, r')$ and get $D(E(c, r'))$. This $E(c, r')$ does not contain any identity information about user u_1, and hence the RS server does not know whose identity that u_2 is trying to verify. At last, RS will check if $D(E(c, r'))$ equals to c. If yes, RS informs VS that the validation is succeed, and no, otherwise. Then VS will notifies u_1 and u_2 the corresponding validation result.

Figure 5 depicts the main messages exchanged during this validation phase. For the above scheme to work, ID_{u_1} needs to have a multiplicative inverse in $\mathbb{Z}_N$. Since $N = pq$, and p and q are very large prime numbers, this requirement can be easily satisfied in our problem domain.

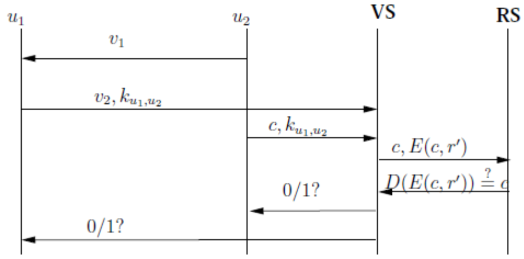

Fig. 5. Ownership Validation

3) Generation of Randomized Authentication ID: In RAU, each randomized authentication ID is only used once so that a user's moving trajectory will not be tracked by any party in the system. To self-generate a new randomized ID, the u_1 just needs to compute it based on Equation 5.

$$\text{RID}^{\text{i}}_{\text{u}_1} = \text{RID}^{\text{i}-1}_{\text{u}_1} * E(0, r^i_{u_1}) \tag{5}$$

Based on the addition property of Homomorphic encryption (Equation 1), the new randomized ID is again the encryption of the real identity which can be deduced as follows:

$$\text{RID}^{\text{i}}_{\text{u}_1} = E(\text{ID}_{\text{u}_1}, r^{i-1}_{u_1}) * E(0, r^i_{u_1}) = E(\text{ID}_{\text{u}_1} + 0, r')$$

It is worth mentioning that by leveraging this addition property, the generation of new randomized ID using the above Equation is more efficient than directly encrypting the real identity again.

The challenge here is how to let the verification server (VS) know that this new ID is valid so that the authentication can be performed later. Recall that the VS maintains a list of valid randomized IDs received from the RS server, but the VS is not able to compute any new ones. Only the RS server knows the real identity of the vehicle but the RS server is not in charge of verification. A straightforward method is to let the user inform the RS server about the new ID and then let the RS server forward it to the VS server, which however will disclose the user's locations to the RS server. Therefore, we propose a synchronization approach that avoids the communication between the user and the RS server during the authentication. The key idea is to let the RS server generate a randomization seed γ_{u_1} and a randomization interval τ_{u_1} for user u_1 at the registration phase. Every τ_{u_1} time, the RS server will be able to generate the same random number as user u_1 did based on the seed γ_{u_1}. Therefore, the RS server can directly compute the randomization ID of u_1 using Equation 5 without communicating with u_1 and without knowing there is an authentication request. In fact, the RS server computes the randomization IDs for all users every τ_{u_1} time and sends these randomly permuted up-to-date random IDs to the VS server. For maximizing security, the old random IDs will be discarded by the RS server.

When the VS server received the new copy of random IDs (randomly permuted by the RS server), it would not be able to link each new ID to its previous version. It is worth noting that the randomization time intervals are not necessary to be the same for all users.

E. Identity Tracing

In some applications, disputes may occur due to various reasons. Sometimes a third-party law enforcement authority may want to know immediately the real identity of a suspect user who is undergoing an authentication. Sometimes there may be a need to discover the authentication history of a suspect user. Thus, we propose both real-time identity tracing and historical identity tracing.

The real-time identity tracing is easy to achieve. The law enforcement authority submits the tracing request that contains the suspect user's randomized authentication ID to either the VS server or the RS server. If the request is received by the VS server, the VS server will forward the suspect user's randomized ID to the RS server. Upon receiving the suspect user's randomized ID, the RS server uses its private key to decrypt the randomized ID and reports the real identity to the law enforcement authority.

In terms of historical identity tracing, the law enforcement authority captured one randomized ID of the suspect user and wants to know the authentication history of the user to figure out the user's behavior in the network. The law enforcement authority sends the randomized ID of the suspect user to both RS and VS server. The RS server maintains a list of authentication history of all users. For example, each user has a list of randomized authentication IDs that have been or are planning to be used. The VS server maintains all valid authentication IDs.

First, the RS server finds a match in a user's list. If there is a match, the list of randomized IDs will be provided to the law enforcement authority who will subsequently send these IDs to VS. The VS will return the authority the real IDs of these randomized IDs. Based on the location of the suspect, the authority may learn where the suspect has been before. To provide this kind of historical tracing, the only thing needs to be changed is that the RS and VS servers need more memory space to store previously used randomized IDs. In addition, when the VS server performs identity validation, it needs to make sure, old IDs cannot be used again. These modifications can be easily incorporated into our current scheme.

F. Credential or Identity Revocation

Identity revocation is very efficient in our system. Once a suspect user is confirmed to be malicious, the RS and the VS server just need to remove this user's randomization ID from their databases. Any subsequent authentication request for this malicious user will fail as no matching record will be found by the server any more.

V. Security Analysis

In this section, we will analyze the security and privacy features of the proposed RAU system.

A. Undiscoverability of the Challenge During the Ownership Validation

Combining with the semantic security of the Paillier encryption function [5] adopted in this paper, we are able to prove it is computationally infeasible for a malicious user to discover any encrypted message c.

Equation 4 presented in Section IV-D2 can be rewritten as $v = (E(x, r_1))^c * E(0, r_2)$. As discussed previously, since we do not know if it is hard to compute the discrete log of $(E(x, r_1))^c$ to discover c, the purpose of multiplying $(E(x, r_1))^c$ with $E(0, r_2)$ is to make the discovery of c from v computationally infeasible without knowing the secret value x. The following claim holds:

Claim 1: Suppose the encryption key size is sufficient large (i.e., N is at least 1024-bit). Without knowing the secret value x, a malicious user is computationally infeasible to discover c from v where $v = (E(x, r_1))^c * E(0, r_2)$.

Our proof follows from the semantic security property of E (the Paillier encryption function). Suppose the claim is not true, then there exists a probabilistic polynomially bounded algorithm A that can discover c from v. Now A can be used to distinguish the two cipher texts $v = (E(x, r_1))^c * E(0, r_2) = E(cx, r)$ and $v' = E(x, r')$ in polynomial time with hundred percent certainty. Thus, we can claim that v and v' are computationally distinguishable according to Definition 1. On the other hand, since E is semantically secure [5], v and v' are computationally indistinguishable. This contradicts that A can distinguish between v and v'. Therefore, the assumption of the existence of A is invalid. We can conclude that the above claim is true and c cannot be discovered by a malicious user without knowing the secret value x. The proposed ownership validation protocol works correctly and securely.

B. Unforgeability

Our authentication protocol guarantees that no one can use the identity that does not belong to him/her. Under the assumptions that the private key is kept securely at the RS server side, the only option left for the attacker to impersonate legitimate users is to exploit their randomized authentication IDs. There are several possible ways for an attacker to obtain a randomized authentication ID of a user. However, we show in the following that the attacker would not be able to use this ID as its own for authentication purpose.

1) The Replay Attack: Although an attacker can obtain another user's valid authentication ID during authentication, the attacker cannot directly use the received authentication ID again since each ID is allowed to be used only once and is

978-1-5090-4327-9/17/$31.00 ©2017 IEEE

discarded after the use by the VS server. If the attacker tries to re-randomize the received ID using a new random number, the resulting ID will not match any valid authentication ID stored in the VS server. This is because the attacker does not know the randomization seed used by the real owner of the ID, and hence the attacker will not be able to generate the same series of randomized IDs that match the real ones.

2) The Man-in-the-Middle Attack: Previous discussion is focused on the used randomized IDs. We now discuss the case when an attacker steals new or never used randomized IDs from a user, the VS or RS server. Since these IDs have not been used by the real owner, the attacker will be able to go through the user authentication phase, but will be caught at the ownership validation phase. This is because the attacker does not know the real identity of the ID owner or the initial seed used in the user registration phase; hence, the attacker cannot discover the random number included in the challenge message (sent by u_2 as discussed in Section IV-D2).

More formally, the above scenario can also be illustrated as the man-in-the-middle attack. The attacker u_m attempts to forward legitimate user u_1's authentication ID to another legitimate user u_2, and vice versa. The attacker aims to prove to u_1 that he is user u_2, and prove to u_2 that he is user u_1. However, such attack will not succeed because our protocol verifies a mutually agreed random number between each pair of users. This random number is unique for each pair of users at each round of generation. Recall that at the beginning of the user authentication (Section IV-D1), user u_1 and the attacker generate a mutually agreed random number $k_{u_1 u_m}$. User u_1 registered this random number at the VS server along with its randomized authentication ID $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u_1}}$.

For anyone who wishes to verify the validity of $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u_1}}$, he/she needs to provide the random number $k_{u_1 u_2}$ to the VS server to prove that he/she is the person who u_1 is currently communicating with. Therefore, even if the attacker tries to present the obtained $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u_1}}$ to u_2, the attacker would not be able to establish a mutually agreed random number with u_2 that is the same as $k_{u_1 u_2}$, as long as u_2 does not collude with u_m (this is the general assumption under the man-in-the-middle attack). Consequently, if u_2 presents $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u_1}}$ and a different random number say $k_{u_m u_2}$ to the VS server for verification, the VS server will easily discover the matching IDs but un-matching random numbers and conclude that there is a man-in-the-middle attack.

C. Full Privacy Preservation

Our authentication protocol provides full privacy preservation in that it guarantees both server-wise and peer-wise privacy for the users in terms of both anonymity and unlinkability. Considering the peer-wise privacy, under the proposed protocol, a user always self-generates a new randomized authentication ID when establishing a new communication session. Since the encryption scheme we adopted is semantically secure [5], it is computationally infeasible for peer users to know the real identity of others and to link different communication sessions or randomized authentication IDs to the same user as long as the size of the encryption key is large enough (such as 1024 bit).

As for the VS server, it does not have the secret key to decrypt the randomized IDs stored in its database, and hence it does not know the real identity of the user who submits authentication request (again here we assume the encryption key size is sufficiently large). Due to the fact that the IDs are randomized, VS cannot link different RIDs to the same user. As for the RS server, since it does not handle any authentication request that contains randomized IDs during the authentication phase, the RS server does not know which user is sending the authentication request. Therefore, our protocol prevents the RS server from tracking the locations of the users.

D. Traceability

Traceability refers to the ability to reveal a user's real identity requested by the law authorities. This is a seemingly conflicting requirement with respect to the privacy preservation goal of our system. We achieve this by proposing the collaborative identity tracing protocol as presented in Section IV-E. The identity tracing protocol is capable of revealing a suspect user's real identity and his/her whole authentication history to the law authorities without violating the privacy of other legitimate users.

E. Preventing Credential Sharing

Credential sharing means that a legitimate user u_1 gives his random ID $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u_1}}$ to another user u_2 so that u_2 may try to communicate with others $\mathrm{RID}^{\mathrm{i}}_{\mathrm{u_1}}$. Our proposed RAU system prevents such credential sharing as long as the legitimate user does not share his personal identifiable information. Since personal identifiable information (such as SSN) would be very sensitive, there would not be enough incentives for a legitimate user to share such sensitive private information even with their friends. By keeping the personal identifiable information secret, user u_2 who obtained u_1's random ID would not be able to pass the ownership validation phase which requires the knowledge of the personal identifiable information.

VI. Experimental Study

It is worth noting that our protocols do not require the vehicles to be equipped with high performance computing equipments. The following hardware specification is used to simulate the servers, but not the hardware carried by vehicles. We implemented the RAU authentication protocol in C language with GMP library, and run the tests on a PC with Intel Xeon CPU X5675 @3.07GHZ x6 and 12GB memory. We evaluate the efficiency of the total authentication process in terms of communication and computation cost. We did not include the transmission and propagation delays since they depend on specific network configurations.

A. Performance of User Registration

The main computation cost involved in the user registration is the generation of the initial randomized ID for the new user. Each randomized ID is 2048 bits. In the experiments, we

978-1-5090-4327-9/17/$31.00 ©2017 IEEE

observed that the randomized ID generation time is less than **1.9ms** per user. In terms of communication cost, besides the standard secure channel establishment, there is only one round of message exchange between the user and the RS server, whereby the user sends his personal identifiable information to the RS server and the RS server sends back the initial random ID to the user.

B. Performance of User Authentication

Being the best among the exiting solutions, the Anonymous Credential (AC) system [32] achieves most criteria for anonymous authentication as our proposed RAU system. However, as discussed in the related work and the security analysis, our proposed RAU is much more secure than AC in face of various attacks. Now, we will also examine the efficiency of the RAU system and the AC system.

For each round of authentication, the AC protocol perform two phases: generate pseudonym and generate credential, while the RAU protocol also consists of two mandatory phases (randomized ID generation and identity validation) and one optional phase (ownership validation). Without the ownership validation phase, our RAU protocol already offers the same functionality as the AC protocol. Thus, we record both the authentication time with and without the ownership validation phase for the RAU protocol. We use RAU to denote the protocol with all three phases, and RAU$^{\alpha}$ to denote the protocol without the ownership validation phase. Table I reports the running time of each phase in all the protocols. Compared to AC, the proposed RAU protocol does not introduce more computational overhead while achieving better security guarantees (as discussed in Section V). Instead, even the three-phase version of the RAU protocol is more efficient than the AC protocol. The performance of the AC protocol is attributed to the more efficient algorithm design based on the Paillier's encryption. In addition, the average latency for a 4G network is about 50ms [43]. By adding the network latency, the overall authentication time is well below the 100ms requirement in VANETs as stated in [44].

TABLE I
USER AUTHENTICATION TIME

Protocol Phase	**AC**	**RAU**	**RAU$^{\alpha}$**
Phase 1	7.1ms	1.95ms	1.95ms
Phase 2	6.2ms	2.05ms	2.05ms
Phase 3 (Ownership validation)	N/A	6.7ms	N/A
Total Authentication Time	**13.3ms**	**10.7ms**	**4ms**

Next, we compare the communication complexity in terms of the amount of messages exchanged throughout the protocol. In AC, there are five rounds of communication and the message complexity is $14l$ bits, where l is the key size. The computation complexity of the total authentication process is 9 exponentiations as the AC is using the zero-knowledge proof. In our RAU protocol, there are total 4 rounds of communication for authentication user u_1 for user u_2, and two rounds of communication for user u_1 to generate another new randomized ID to be used for the future. The message complexity is $9l + 512$ bits (l is the key size), but the computation only takes one exponentiation. Moreover, the first four steps already complete the authentication for u_1 with respect to u_2. The last two steps can be performed anytime before user u_1 wants to communicate with another user. Table II summarizes the comparison results, whereby RAU$^{\beta}$ denotes the protocol with only the first 4 phases.

TABLE II
COMMUNICATION COMPLEXITY

Protocol	**AC**	**RAU$^{\alpha}$**	**RAU$^{\beta}$**
Communication rounds	5	6	4
Message size	$14 \cdot l$	$9 \cdot l + 512$	$7 \cdot l + 512$
Exponentiations	**9**	**1**	**1**

Please note that for the ownership validation phase, the communication complexity is 5 steps with $8l$ bits message complexity, and the computation cost is 2 exponentiations. Without the ownership validation phase, our protocol offers the same functionality as the AC protocol. Thus, to have a fair comparison, Table II does not include the complexity of the ownership validation phase. In addition, as proved in [32], [33], the AC protocol is only statistically secure. As a result, our protocol is more secure because of the semantic security guarantee of the Paillier encryption scheme.

C. Performance of Identity Tracing

Our proposed RAU protocol has a nice feature of providing user tracing in terms of any dispute. There are two types of tracing available in the RAU system: (i) the real-time identity tracing; and (ii) the historical identity tracing. Table III reports the running time of tracing a single user. It is not surprising to see that the real-time identity tracing is much more efficient than the historical identity tracing. This is because the real-time identity tracing only needs to recover a single user ID whereas the historical identity tracing needs to check the disputed randomized ID against a list of randomized IDs which have been used by the same user in the past.

TABLE III
IDENTITY TRACING TIME

Protocol	**Real-time Tracing**	**Historical Tracing**
RAU	1.8ms	3.49s

VII. CONCLUSION AND FUTURE WORK

In this paper, we present an anonymous authentication system RAU for VANET users. The proposed protocols leverage the properties of a semantically secure public-key additive homomorphic encryption scheme. The proposed RAU system overcomes shortcomings in other existing works, and achieves a set of desired properties including unforgeability, full privacy preservation, identity tracing and being secure against various types of attacks. In particular, it is more efficient and secure than the widely used anonymous credential system proposed in [32], [33]. In the future, we plan to integrate the proposed authentication protocol into routing protocols to build a comprehensive system on a VANET simulation platform.

References

[1] M. Gerla and M. Gruteser, “Vehicular networks: Applications, protocols, and testbeds,” *In Emerging Wireless Technologies and the Future Mobile Internet*, pp. 201–241, 2011.

[2] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, and W. Trappe, “Parknet: drive-by sensing of road-side parking statistics,” in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys 2010)*, 2010, pp. 123–136.

[3] W. Viriyasitavat, F. Bai, and O. K. Tonguz, “Toward End-to-end Control in VANETs,” in *IEEEVehicular Networking Conference (VNC)*, 2011, pp. 78–85.

[4] D. Lin, E. Bertino, R. Cheng, and S. Prabhakar, “Location privacy in moving-object environments,” *Transactions on Data Privacy*, vol. 2, no. 1, pp. 21–46, 2009.

[5] P. Paillier, “Public key cryptosystems based on composite degree residuosity classes,” in *Advances in Cryptology - Eurocrypt ’99 Proceedings, LNCS 1592*. Prague, Czech Republic: Springer-Verlag, 1999, pp. 223–238.

[6] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen, “Ecpp:efficient conditional privacy preservation protocol for secure vehicular communications,” in *Proc. of IEEE Conference on Computer Communications*, 2008, pp. 1229 – 1237.

[7] A. Squicciarini, D. Lin, and A. Mancarella, “Paim: Peer-based automobile identity management in vehicular ad-hoc network,” in *Proc. of the IEEE Computer Software and Applications Conference (COMPSAC)*, 2011.

[8] C. Zhang, R. Lu, X. Lin, P. Ho, and X. Shen, “An efficient identity-based batch verification scheme for vehicular sensor networks,” in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*. IEEE, 2008, pp. 246–250.

[9] M. Raya and J. P. Hubaux, “Securing vehicular ad hoc networks,” in *Journal of Computer Security*, 2007, pp. 39–68.

[10] K.-A. Shim, “Cpas: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks,” in *IEEE Transaction on Vehicular Technology*, 2012, pp. 1874–1883.

[11] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, and J.-P. Hubaux, “Certificate revocation in vehicular networks,” *Laboratory for computer Communications and Applications (LCA) School of Computer and Communication Sciences, EPFL, Switzerland*, 2006.

[12] X. Sun, “Anonymous, secure and efficient vehicular communications,” Master’s thesis, The University of Waterloo, Waterloo, Ontario, Canada, 2007.

[13] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, and A. Lioy, “Efficient and robust pseudonymous authentication in vanet,” in *Proceedings of the fourth ACM international workshop on Vehicular ad hoc networks*. ACM, 2007, pp. 19–28.

[14] Y. Sun, R. Lu, X. Lin, X. Shen, and J. Su, “An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications,” *Vehicular Technology, IEEE Transactions on*, vol. 59, no. 7, pp. 3589–3603, 2010.

[15] A. Studer, E. Shi, F. Bai, and A. Perrig, “Tacking together efficient authentication, revocation, and privacy in VANETs,” in *Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks*, ser. SECON’09. Piscataway, NJ, USA: IEEE Press, 2009, pp. 484–492.

[16] J. Sun, C. Zhang, Y. Zhang, and Y. Fang, “An identity-based security system for user privacy in vehicular ad hoc networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 21(9), pp. 1227–1239, 2010.

[17] C. Zhang, P.-H. Ho, and J. Tapolcai, “On batch verification with group testing for vehicular communications,” *Wireless Networks*, vol. 17(8), pp. 1851–1865, 2011.

[18] J. Zhang, Y. Cui, and Z. Chen, “Spa: Self-certified pkc-based privacy-preserving authentication protocol for vehicular ad hoc networks.”

[19] J. Li, H. Lu, and M. Guizani, “Acpn: a novel authentication framework with conditional privacy-preservation and non-repudiation for vanets,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 4, pp. 938–948, 2015.

[20] K. Sha, Y. Xi, W. Shi, L. Schwiebert, and T. Zhang, “Adaptive privacy-preserving authentication in vehicular networks,” in *Communications and Networking in China, 2006. ChinaCom’06. First International Conference on*. IEEE, 2006, pp. 1–8.

[21] C. D. Jung, C. Sur, Y. Park, and K.-H. Rhee, “A robust conditional privacy-preserving authentication protocol in vanet,” *Social Informatics and Telecommunications Engineering*, vol. 17, pp. 35–45, 2009.

[22] Y. Wang, H. Zhong, Y. Xu, and J. Cui, “Ecpb: Efficient conditional privacy-preserving authentication scheme supporting batch verification for vanets,” *International Journal of Network Security*, vol. 18, no. 2, pp. 374–382, 2016.

[23] D. Chaum and E. V. Heijst, “Group signatures,” in *Advanced CryptologyEurocryptS*, 1991, pp. 257–265.

[24] Y. Hao, C. Yu, C. Zhou, and W. Song, “A distributed key management framework with cooperative message authentication in vanets,” *Selected Areas in Communications, IEEE Journal on*, vol. 29, no. 3, pp. 616–629, 2011.

[25] F. Wang, Y. Xu, H. Zhang, Y. Zhang, and L. Zhu, “2flip: A two-factor lightweight privacy-preserving authentication scheme for vanet,” *IEEE Transactions on Vehicular Technology*, vol. 65, no. 2, pp. 896–911, 2016.

[26] L. Yeh, Y. Chen, and J. Huang, “Paacp: A portable privacy-preserving authentication and access control protocol in vehicular ad hoc networks,” *Computer Communications*, vol. 34, no. 3, pp. 447–456, 2011.

[27] Z. Tan, “A lightweight conditional privacy-preserving authentication and access control scheme for pervasive computing environments,” *Journal of Network and Computer Applications*, 2012.

[28] C. Gamage, B. Gras, B. Crispo, and A. Tanenbaum, “An identity-based ring signature scheme with enhanced privacy,” in *Securecomm and Workshops*, 2006, pp. 1–5.

[29] S. Zeng, Y. Huang, and X. Liu, “Privacy-preserving communication for vanets with conditionally anonymous ring signature,” *International Journal of Network Security*, vol. 17, no. 2, pp. 135–141, 2015.

[30] D. Chaum, “Blind signatures for untraceable payments,” in *Advanced CryptologyEurocryptS*, 1982, p. 199C203.

[31] U. Rajput, F. Abbas, J. Wang, H. Eun, and H. Oh, “Cacppa: A cloud-assisted conditional privacy preserving authentication protocol for vanet,” in *2016 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid)*, 2016, pp. 434–442.

[32] J. Camenisch and E. V. Herreweghen, “Design and implementation of the *idemix* anonymous credential system,” in *ACM Conference on Computer and Communications Security*, 2002, pp. 21–30.

[33] J. Camenisch and A. Lysyanskaya, “An efficient system for non-transferable anonymous credentials with optional anonymity revocation,” in *EUROCRYPT*, 2001, pp. 93–118.

[34] J. Camenisch and E. Van Herreweghen, “Design and implementation of the idemix anonymous credential system,” in *Proceedings of the 9th ACM conference on Computer and communications security*. ACM, 2002, pp. 21–30.

[35] J. Camenisch and T. Groß, “Efficient attributes for anonymous credentials,” in *Proceedings of the 15th ACM conference on Computer and communications security*. ACM, 2008, pp. 345–356.

[36] E. Brickell, J. Camenisch, and L. Chen, “Direct anonymous attestation,” in *Proceedings of the 11th ACM conference on Computer and communications security*. ACM, 2004, pp. 132–145.

[37] E. Cesena, H. Löhr, G. Ramunno, A.-R. Sadeghi, and D. Vernizzi, “Anonymous authentication with tls and daa,” in *Trust and Trustworthy Computing*. Springer, 2010, pp. 47–62.

[38] C. Wachsmann, L. Chen, K. Dietrich, H. Löhr, A.-R. Sadeghi, and J. Winter, “Lightweight anonymous authentication with tls and daa for embedded mobile devices,” in *Information Security*. Springer, 2011, pp. 84–98.

[39] J. Camenisch and A. Lysyanskaya, “Dynamic accumulators and application to efficient revocation of anonymous credentials,” in *Advances in CryptologyłCRYPTO 2002*. Springer, 2002, pp. 61–76.

[40] S. Goldwasser, S. Micali, and C. Rackoff, “The knowledge complexity of interactive proof systems,” in *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, Providence, Rhode Island, U.S.A., 1985, pp. 291–304.

[41] O. Goldreich, “The foundations of cryptography: Encryption schemes,” *IEEE Transactions on Vehicular Technology*, vol. 2, 2004.

[42] W. Diffie and M. Hellman, “New directions in cryptography,” *IEEE*, vol. IT-22, no. 6, pp. 644–654, 1976.

[43] . m. L. 5G Vision: 100 Billion connections and . G. Throughput, *http://www.huawei.com/minisite/5g/en/defining-5g.html*, 2016.

[44] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, and T. Weil, “Vehicular networking: A survey and tutorial on requirements,architectures, challenges, standards and solutions,” *IEEE Communications Surveys & Tutorials*, vol. 13, no. 4, pp. 584–616, 2011.

978-1-5090-4327-9/17/$31.00 ©2017 IEEE