Pervasive and Mobile Computing 44 (2018) 31–44

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

# Highly efficient randomized authentication in VANETs

Jian Kang [a], Dan Lin [a,*], Wei Jiang [a], Elisa Bertino [b]

[a] Department of Computer Science, Missouri University of Science and Technology, United States
[b] Department of Computer Science, Purdue University, United States

ARTICLE INFO

*Article history:*
Available online 6 February 2018

*Keywords:*
VANET
Authentication
Traceability
Privacy

ABSTRACT

Considering the huge number of vehicles on the roads, Vehicular Ad-hoc Networks (VANETs) are envisioned to foster a variety of new applications ranging from road safety enhancement to mobile entertainment. These new VANET applications all face a critical challenge which is to ensure the identity and location privacy of vehicles' owners who participate in such ad-hoc network. In this paper, we propose a highly efficient 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.



## 1. 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 components 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.

* Corresponding author.
*E-mail addresses:* jkb7c@mst.edu (J. Kang), lindan@mst.edu (D. Lin), wjiang@mst.edu (W. Jiang), bertino@purdue.edu (E. Bertino).

https://doi.org/10.1016/j.pmcj.2018.01.004
1574-1192/

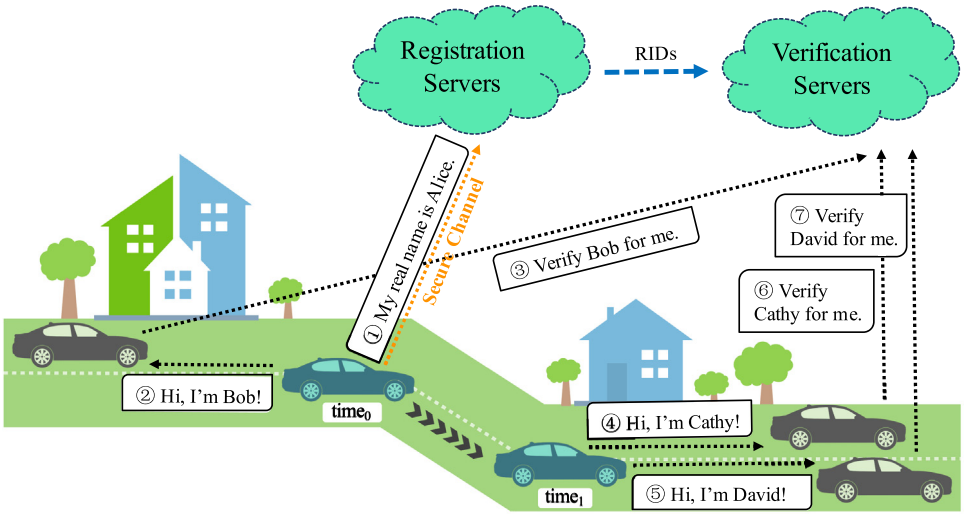



**Fig. 1.** Example scenarios of the proposed randomized authentication in VANETs. (1) At $time_0$, the blue car first registered her real identity as 'Alice' at the registration server. (2) The blue car self-generates a new randomized ID (RID) 'Bob' to communicate with the black car on the left. (3) The black car asks the verification server to verify if 'Bob' is a legitimate user. Here the verification server alone will only know that 'Bob' is a legitimate user but does not know 'Bob' is 'Alice' and hence cannot track 'Alice'. (4) At $time_1$, the blue car moves to another street, and uses another new ID 'Cathy' and 'David' to communicate with the two black cars on the right side of the figure, respectively. (5) These two black cars request the verification server to verify 'Cathy' and 'David'. Again, the verification server would not know 'Cathy' and 'David' are actually 'Alice'. Both registration server and verification server cannot track the blue car's movement. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

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 higher mobility and the communication links in VANETs breaks more frequently than in MANETs [5]. 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 propose a highly efficient authentication protocol RAU$^+$ based on our prior work, i.e., Randomized AUthentication (RAU) [6]. The RAU$^+$ inherits all the security properties from RAU, i.e., preserves vehicles' privacy while ensuring traceability. In particular, the RAU$^+$ leverages homomorphic encryption and enables individual vehicles 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 requester. Fig. 1 shows simple example scenarios. 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 RAU, the RAU$^+$ is more advanced in terms of efficiency and usability. Specifically, the RAU$^+$ provides a new type of authentication, namely aggregated authentication, which allows one vehicle to verify multiple vehicles simultaneously with a single request message to the verification server. In RAU, such verification with multiple vehicles will have to be conducted separately for each vehicle and hence these requests are very time consuming. Compared to existing VANET authentication works [7–11], 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 efficient which meets the real-time constraints in VANET applications well. A 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 2 reviews related works and Section 3 introduces preliminaries. Section 4 presents the proposed privacy-preserving randomized authentication protocol. Section 5 conducts security analysis. Section 6 reports the experimental results. Finally, Section 7 concludes the paper.

## 2. 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 [10]. 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 [12] proposed efficient revocation schemes. However, these schemes violate the location privacy requirement and are subject to a movement tracking attack. 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 [9,11,15–19], achieving different degrees of improvement over the key revocation problem. However, in most of 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 protocols is group-based [7,20–22]. 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. 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. [7], RSUs (Road Side Units) 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. [23] and the decentralized certificate authority with the biological-password-based two-factor authentication by Wang et al. [24]. Besides group-based signature schemes, other techniques have also been proposed to achieve anonymity within a group. For example, Zhang et al. [9] 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 [8], 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 anonymous authentication of vehicles. However, the proposed protocol requires a complicated group management strategy which introduces extra overhead to the system. In addition, there have been some works [25–28] developed based on the ring signature or blind signature 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 to 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 [29] 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 [30–33] which has been chosen as a comparison approach in our experiments due to its popularity and the similar security goal of providing randomized identities for users. However, the AC system has the following limitations that make it not the best fit for VANETs. First, the AC system uses zero-knowledge proofs to verify if a user possesses a valid credential. Zero-knowledge proofs are computational expensive which may not be practical in VANETs since vehicles may have already moved a far distance when waiting for authentication. Second, the AC system provides a authentication protocol between users and organizations and hence assumes that they are already connected by pre-established secure channels. However, establishing security channels between vehicles is a tricky task and could also be time consuming if it is not integrated with the authentication protocol. 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 discourage users from sharing his credentials, one solution is to ask each user to give the organizations (or verifiers) a verifiable encryption of his valuable secret information that can be decrypted with his secret key. Unfortunately, this approach is not practical either in the scenario of vehicle-to-vehicle communication because vehicles are not trustworthy organizations and hence servers need to assist the process which could be very time consuming and requires modification of the original AC protocols. Although there is another solution to avoid credential sharing by using hardware [34–36], our proposed approach does 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 [37], 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 are some other extensions of AC [32,33], but none of them directly addresses the aforementioned disadvantages.

## 3. Preliminary

For a better understanding, we first briefly review the additive homomorphic probabilistic public key encryption ($\text{HEnc}^+$) 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 $\text{HEnc}^+$ 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 $\text{HEnc}^+$ 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 $x_1 + x_2$.

$$E(x_1) * E(x_2) = E(x_1 + x_2) \tag{1}$$

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

$$E(x)^c = E(c * x) \tag{2}$$

- The encryption function has semantic security as defined in [38], 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 [39] 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, the size of $N$ should be at least 1024 bits. The encryption function is defined as follows for $x$:

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

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 $r_2$.

## 4. RAU$^+$: advanced randomized authentication system

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

### 4.1. 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. At the beginning, users register at the RS server. The RS server shares an initial randomized authentication ID and secret key of each user with the VS server. Whenever a user wants to communicate with other vehicles, he/she can randomly generate pseudo identities and secret keys which can be verified by the VS server to prove the user's ownership of pseudonym. If there is any dispute, the two servers will conduct a tracing protocol to figure out the real identity of a malicious user. It is worth noting that our protocols ensure that all communications between vehicles and registration servers in VANETs are via secure channels. The establishment of secure channels are integrated with the registration protocol. The fundamental technology adopted is Paillier encryption scheme [39]. The detailed steps in each phase are elaborated in the following subsections.

### 4.2. Threat model

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

- Like all existing work discussed in Section 2, we assume that the two authentication servers adopted in 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 no financial incentive to collude to damage their reputations.

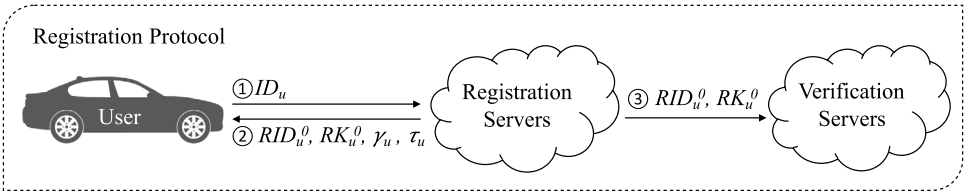



**Fig. 2.** User registration. (Please note that we show only the message content here. All messages are in fact encrypted.)

- The users can be malicious. A malicious user can impersonate another user. When the users are malicious, we will consider three common attacks under most authentication systems: man-in-the-middle, replay and credential sharing.

### 4.3. User registration

The registration phase is for a vehicle (i.e., VANET user) to be authenticated by the server and obtain an initial random ID ($RID_u^0$), a random secret key ($RK_u^0$), a randomization seed $\gamma_u$ and a randomization interval $\tau_u$, based on which the user would be able to self-generate any number of random IDs and keys to communicate with other peer vehicles later on. The registration protocol is illustrated in Fig. 2.

As a one-time setup, the registration server (RS) generates its own public–private key pair using the Paillier encryption scheme [39], and the public key is known by all entities in the VANETs. 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 ($RID_u^0$) and secret key ($RK_u^0$) for user $u$ as follows:

$$RID_u^0 = E(ID_u, r_u^0) \tag{3}$$

$$RK_u^0 = E(RK_u^b, r_u^0) \tag{4}$$

where $E(X, r)$ is a Paillier encryption of $X$ with a random number $r$ using the RS' public key, and $RK_u^b$ is a basic secret key randomly generated by RS server. $RID_u^0$ and $RK_u^0$ are 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 $RID_u^0$ and random secret key $RK_u^0$ in their local databases. 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.

### 4.4. User authentication

We now proceed to present the two-way authentication protocol for user $i$ and $j$ to verify each other's legitimacy. The authentication protocol consists of two main phases: (i) identity validation and (ii) generation of a new randomized authentication ID and secret key. If needed, user $i$ can perform concurrent authentication sessions with multiple users by doing aggregated identity validation to reduce the communication overhead in the VANET and the workload of the verification server.

#### 4.4.1. Single Identity Validation

After the registration, the user $i$ obtained one initial randomized ID $RID_u^0$ and secret key $RK_u^0$ from the registration server. User $i$ can use them directly for the follow-up communication with other vehicles, or generate other new randomized IDs and secret keys based on this initial randomized ID and secret key using the protocol in Section 4.4.3. Let $RID_i$ denote the randomized ID that user $i$ will use to authenticate himself with user $j$. The following steps will be performed (illustrated in Fig. 3):

- **Generating random numbers**
  User $i$ first generates a random number $r_i$ for replay and man-in-the-middle attacks prevention purpose (discussed in Section 5). For two-way authentication, after receiving user $i$'s randomized ID, user $j$ will generates the random number $r_j$ as well.

[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.

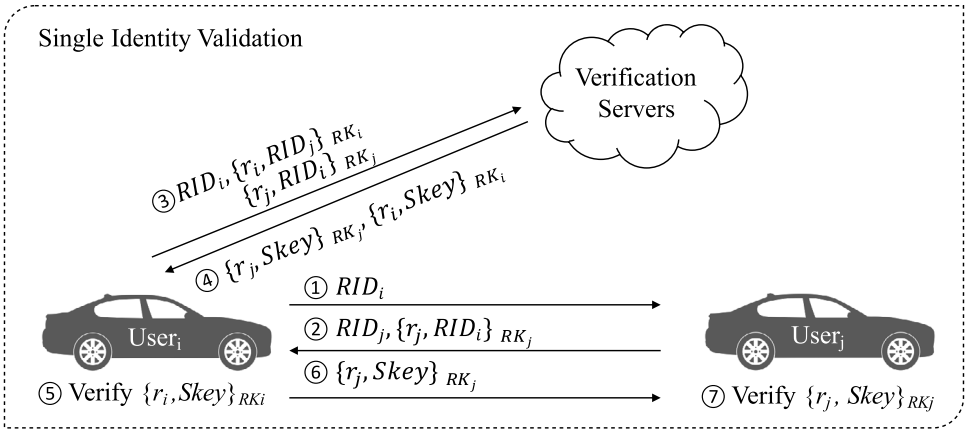



**Fig. 3.** Single identity validation.

- **Exchanging the randomized ID and generating verification data**
  Before sending the verification data to the VS server, user $i$ first sends its $RID_i$ to user $j$. Then, user $j$ reply his/her $RID_j$ to user $i$. Besides, user $j$ encrypts his/her verification data $E_j = \langle r_j, RID_i \rangle$ by $RK_j$ and sends it to user $i$. After receiving this reply message, user $i$ generates his/her encrypted verification data $E_i = \langle r_i, RID_j \rangle$ by $RK_i$ as well.
- **Verifying the randomized ID and secret key**
  After exchanging the randomized IDs, user $i$ sends $RID_i$, $E_i$ and $E_j$ to the VS server. The VS server will first check whether $RID_i$ is exists in its database and still effective. Then, by decrypting $E_i$ using $RK_i$ and decrypting $E_j$ using $RK_j$, the VS server verifies that $E_i$, $E_j$ are generated by user $i$ and $j$ respectively. By further comparing the $RID_i$ in $E_j$ and $RID_j$ in $E_i$ in the verification data, the VS server can verify that both user $i$ and user $j$ are willing to authenticate each other's identity. Finally, the VS server verifies that $RID_j$ is exists in its database and still effective as well.
- **Vehicle-side verification and secure channel establishment**
  Once the VS server verifies user $i$ and user $j$'s verification data, it will randomly generates a session key $Skey$, and sends the encrypted response data $R_i = \langle Skey, r_i \rangle$ (encrypted by $RK_i$) and $R_j = \langle Skey, r_j \rangle$ (encrypted by $RK_j$) to user $i$. User $i$ will then forward $R_j$ to user $j$. Both user $i$ and user $j$ decrypt $R$ using their own $RK$ and check if the random number is correct to avoid the message replay attack. After that, the secure channel between user $i$ and $j$ can be established by using the session key $Skey$. Any attacker cannot obtain this session key because they do not know $RK_i$ and $RK_j$.

It is worth noting that the RAU$^+$ protocols naturally support identity ownership verification to prevent identity sharing, which will be discussed in Section 5.

*4.4.2. Aggregated identity validation*

The single identity validation protocol deals with a pair of vehicles' authentication each time. In some cases, we may need a more efficient method to authenticate a group of vehicles. For example, in a safety enhancement application, when a vehicle $i$ moves towards an intersection, it may need to know the speed and moving direction of vehicles approaching the intersection. At this point, vehicle $i$ needs to authenticate itself to the vehicles near the same intersection, and these vehicles may need to verify vehicle $i$ as well. If using the single identity validation protocol, a large number of verification requests may jam the VS server. Therefore, we propose an aggregated identity validation protocol for a vehicle to authenticate with multiple surrounding vehicles simultaneously. This protocol can reduce both the number of connection establishments and the total transferred bytes between users and the VS server.

To conduct the aggregated identity validation, vehicle $i$ first sends a message to communicate with the group of vehicles that would like to verify its identity. After receiving the verification data from them, vehicle $i$ aggregates the information and sends it to the VS server for verification. Finally, vehicle $i$ forwards the verification results received from the VS server to the group of the vehicles. Fig. 4 illustrates the protocol:

- **Generating verification data**
  User $i$ first generates a random number $r_i$, then sends $RID_i$ to users $j, k, \ldots, z$. User $x$ who receives this request generates encrypted verification data $E_x$ and sends it back to user $i$.
- **Aggregating and submitting verification data**
  After receives $RID$s and verification datas $E_j, E_k, \ldots, E_z$, user $i$: (1) generates the aggregated verification data $AE_i = \langle r_i, RID_j, RID_k, \ldots, RID_z \rangle$ and encrypts it using $RK_i$, and (2) sends $RID_i, AE_i, E_j, E_k, \ldots, E_z$ to VS for verification. Considering that some responses may not be able to transfer to user $i$ on time due to network delay or other reasons, after

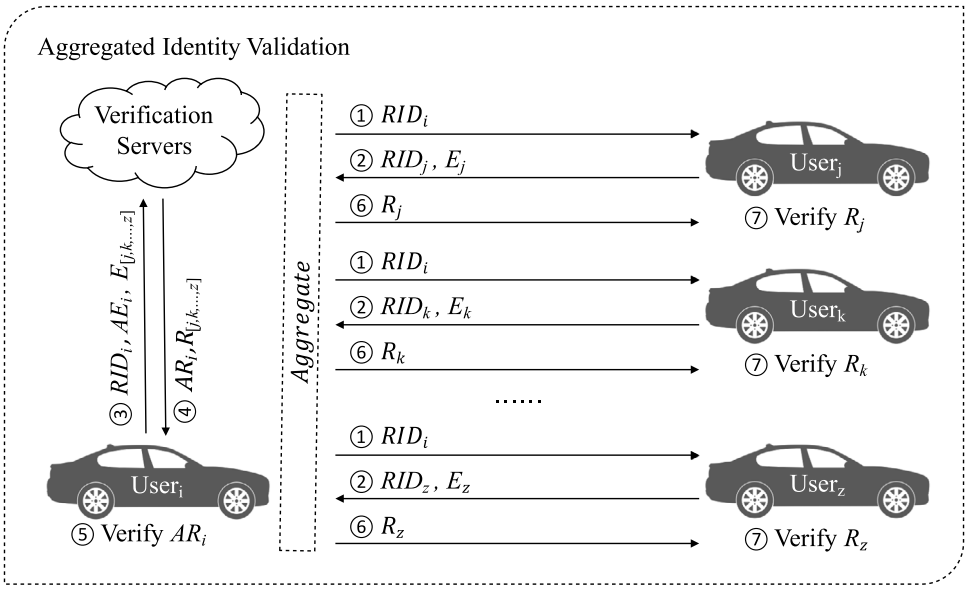



**Fig. 4.** Aggregated identity validation.

sends out request messages, user $i$ will wait $T_{w1}$ to collect, aggregate and send out verification datas. Then, he/she will wait another $T_{w2}$ to deal with other delayed responses and sends out verification datas to the VS server one more time. After that, user $i$ will cancel all the remaining authentication processes.

- **Verifying the randomized IDs and secret keys**
  Upon receiving the verification datas, the VS server firstly decrypts $AE_i$ using $RK_i$ to check that $AE_i$ is actually generated by user $i$. Then, it will check whether $RID_i$, $RID_j$, ..., $RID_z$ are exist in its database and still effective. After that, the VS server will decrypts $E_j$, $E_k$, ..., $E_z$, verifies that those encrypted verification data are generated by correct $RK$. If the verification succeeds, the VS server will generates the session key $Skey_x$ for channel between user $i$ and $x$, encrypts the verification results $R_x = \langle r_x, Skey_x \rangle$ by $RK_x$, and then sends them back to user $i$. Besides, the VS generates aggregated verification result $AR_i = \langle r_i, Skey_j, Skey_k, \ldots, Skey_z \rangle$, encrypts it by $RK_i$ and sends it to user $i$ as well.
- **Vehicle-side verification and secure channel establishment**
  Once the VS server sends back the verification results, user $i$ first keeps $AR_i$, and then forwards $R_j$, $R_k$, ..., $R_z$ to users $j$, $k$, ..., $z$, respectively. After that, each user $x$ decrypts the verification result using his/her own random secret key $RK_x$ and verifies the random number $r_x$. By doing that, user $x$ can verifies that the result is generated by the VS server and obtains the session key $Skey_x$ from the decrypted messages to establish the secure channel between user $i$ and $x$.

#### 4.4.3. *Generation of randomized authentication ID and secret key*

In RAU$^+$, each randomized authentication ID is only used once or in a short duration so that a user's moving trajectory will not be tracked by any party in the system. To self-generate the $k$th new randomized ID and secret key, the user $i$ can compute them using Eqs. (5) and (6).

$$RID_i^k = RID_i^{k-1} * E(0, r_i^k) \tag{5}$$

$$RK_i^k = RK_i^{k-1} * E(0, r_i^k). \tag{6}$$

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

$$RID_i^k = E(ID_i, r_i^{k-1}) * E(0, r_i^k) = E(ID_i + 0, r').$$

It is worth mentioning that by leveraging this addition property, the generation of the new randomized ID 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. 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 $\gamma_i$ and a randomization interval $\tau_i$ for user $i$ at the registration phase. Every $\tau_i$ time, the RS server will be able to generate the same random number as user $i$ did based on the seed $\gamma_i$. Therefore, the RS server can directly compute the randomization ID and secret key of $i$ using Eq. (5) without communicating with user $i$ and without knowing there is an authentication request. After that, RS sends these randomly permuted up-to-date random ID and key to the VS server. To further enhance security, the old random IDs and keys are discarded by the RS server.

When the VS server received the new copy of the random IDs and keys (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, and each time the RS server generates the same random ID and secret key as user $i$ did based on the seed $\gamma_i$, both RS server and user $i$ can generate a same new $\tau_i$ as well so that the attacker cannot associate different *RID*s with their time intervals. Since it is hard for users synchronize their local clock with RS and VS perfectly, there are some special designs in our protocol to overcome this problem: (1) After received by VS, every randomized ID will be discarded after a period of time $T_d$ ($T_d$ is a pre-defined time period that for every user $i$, $T_d$ always larger than $\tau_i$). (2) Users will resynchronize their local clock after every Registration/Authentication phase and adjust their local clock $T_d/2$ slower than RS/VS. These two designs ensure that as long as the time difference between servers and users is less than $T_d/2$, the RID will be validated successfully.

### 4.5. 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 planned 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 which will subsequently send these IDs to VS. The VS will return the authority the verification records of these randomized IDs. Based on the location of the suspect, the authority may learn when and 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 and verification records. 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.

### 4.6. Credential or identity revocation

Identity revocation is very efficient in our system. Once a suspicious user is confirmed to be malicious, the RS and the VS servers just need to remove this user's randomized ID and secret key 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.

### 4.7. Non-transferability

The RAU$^+$ system can prevent users from sharing their credentials with other users without any additional workload. If user $i$ wants to lend his credential to user $j$, he has to share the $RID_i$ and $RK_i$ with user $j$. If any user requests a random number $r$ from RS, encrypts $r$ by $RK_i$ and sends them back to RS, the RS will return the user's real identity and other valuable information encrypted by $RK_i$ as well. Because of that, if user $i$ shares his credential with user $j$, user $j$ will be able to use all of user $i$'s credential and obtain user $i$'s valuable information. This effectively discourages user $i$ from sharing his credential. Besides, since the VS does not know the randomization seed and interval, it is impossible for VS to get user's valuable information.

## 5. Security analysis

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

### 5.1. 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.

#### 5.1.1. The replay attack

Although an attacker may obtain another user's valid authentication ID during authentication, the attacker cannot directly use the received authentication ID again since each ID is associated with a random secure key $RK$. Without knowing this secure key, the attacker cannot obtain the session key and finish the authentication successfully. The encrypted verification data also cannot be replayed because it contains the random number which is only knowing by the user. If any attacker replays this data, he will not be able to decrypt the session key in the response data from VS or pass the user side verification of the random number.

#### 5.1.2. The man-in-the-middle attack

We now discuss the case when an attacker attempts to forward a new (or never used) randomized IDs from user $i$ to user $j$ whereby the attacker tries to pretend to be user $i$. Even though these IDs have not been used by the real owner, the attacker will not be able to pass the user authentication phase. This is because the attacker does not know the random secret key $RK_i$ belonging to the true ID owner–user $i$; hence, the attacker cannot encrypt the validation data using the correct secret key (step 3 in Fig. 3). As a result, the VS server will not be able to decrypt the authentication request using the correct $RK_i$ either, and hence will reject this authentication.

### 5.2. 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 [39], 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 *RID*s, the *RK*s and the renew times of them are randomized, and all the *RID*s have the same effective duration, VS cannot link different RIDs and RKs 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.

### 5.3. Prevention of credential sharing

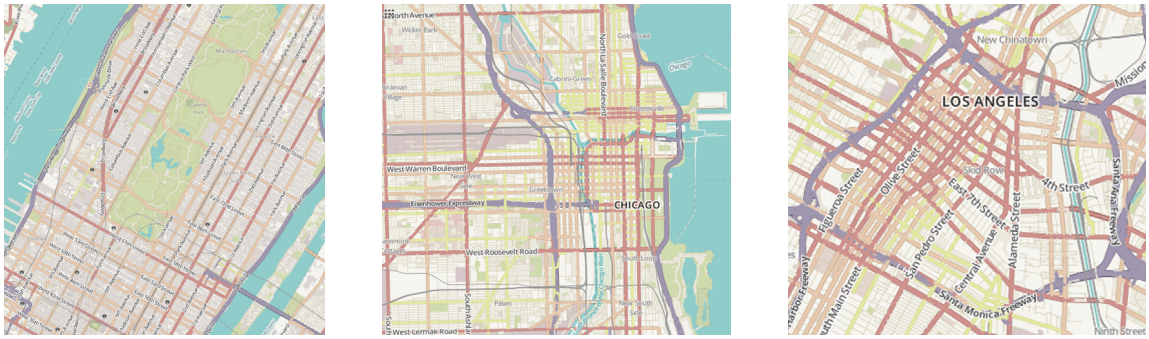
Credential sharing means that a legitimate user $i$ gives his random ID $RID_i$ to another user $j$ so that user $j$ may try to communicate with others using $RID_i$. Our proposed RAU$^+$ system prevents such credential sharing as long as the legitimate user does not share his personal identifiable information which is encrypted by $RK_i$. 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 secret key $RK$ and personal identifiable information secret, user $j$ who obtained $i$'s random ID would not be able to pass the validation phase which requires the knowledge of the randomized secret key $RK$.

### 5.4. 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 4.5. 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.

## 6. Experimental study

In this section, we first introduce the experimental settings and then report the experimental results.




(a) Manhattan. (b) Chicago. (c) Los Angeles.

**Fig. 5.** Real maps used in the simulations.

### 6.1. Experimental settings

We compare our proposed RAU$^+$ with two existing approaches: (i) the RAU protocol which is our prior work [6] on randomized authentication; (ii) Anonymous Credential (AC) system [31] which is the most related work with similar security goals to our work.

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 whole system including network simulation, vehicles movement simulation and server/vehicle side computing simulation, but not the hardware carried by vehicles. We implemented the RAU$^+$ authentication protocol in Java and C++ languages and run the tests on a PC with Intel Core i7 CPU 2.6 Ghz and 16 GB memory. We evaluate the efficiency of the total authentication process in terms of communication and computational cost. The transmission and propagation delays were simulated as well.

The authentication protocols are implemented by NetBeans with JDK 8 to evaluate the time performance on the vehicle side and the server side. We use the vehicular mobility simulator SUMO (v0.23.0) to simulate the vehicles' movements in three real maps as shown in Fig. 5: Manhattan (4.5 km $*$ 5.5 km), Chicago (6 km $*$ 7 km) and Los Angeles (5 km $*$ 4.5 km).

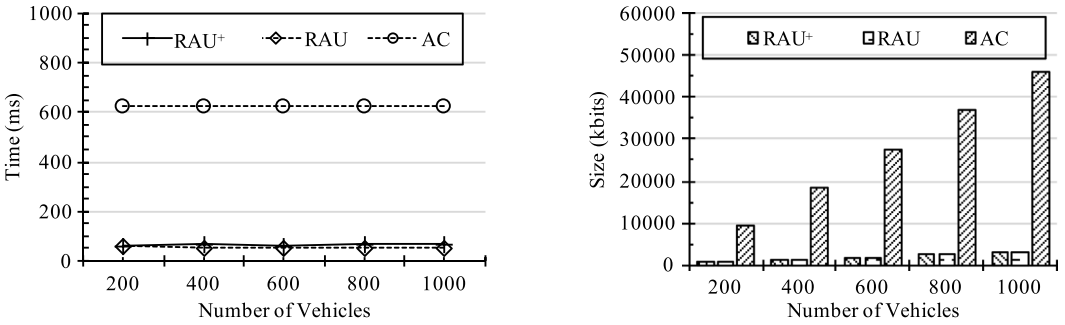
The number of vehicles is ranging from 200 to 1000. Unless noted, we use the Manhattan map and set the number of vehicles to 800. The speed of vehicles is 30 miles per hour inside the city, and 60 miles per hour on the highway. In the SUMO simulation, vehicles will slow down when approaching an intersection and stop if there is a traffic jam. The simulation of network is conducted by the Network Simulator NS-3 (v3.26). The maximum transmission range in NS-3 is set to 100 m and the transmission rate of the wireless channel among vehicles is 6 Mbps. The network delay between vehicles and server is 20 ms. The VS and RS are connected through 100 Mbps wired network. The total simulation time is 120 s. Firstly, the simulation runs for 15 s to insert all vehicles. Then, vehicles begin the registration phase. At a random time after the 60th seconds, each vehicle selects several nearby vehicles within 80 m to start the aggregated authentication process. The default value of the maximum number of the selected vehicles is 10.

### 6.2. Experimental results

In all approaches, each vehicle performs one registration process and one group authentication operation whereby the vehicle randomly selects several nearby vehicles to start a two-way authentication between himself and other vehicles. For example, for vehicle *i*, it first starts the registration phase at 8th second, then chose 10 nearby vehicles to proves that it is a legitimate user. Those 10 vehicles also prove to vehicle *i* that they are legitimate users. The operations occur at random times chosen from a same random sequence so that in the simulations of the three approaches, each vehicle executes the same operation at the same time. In this way, we ensure a fair comparison.

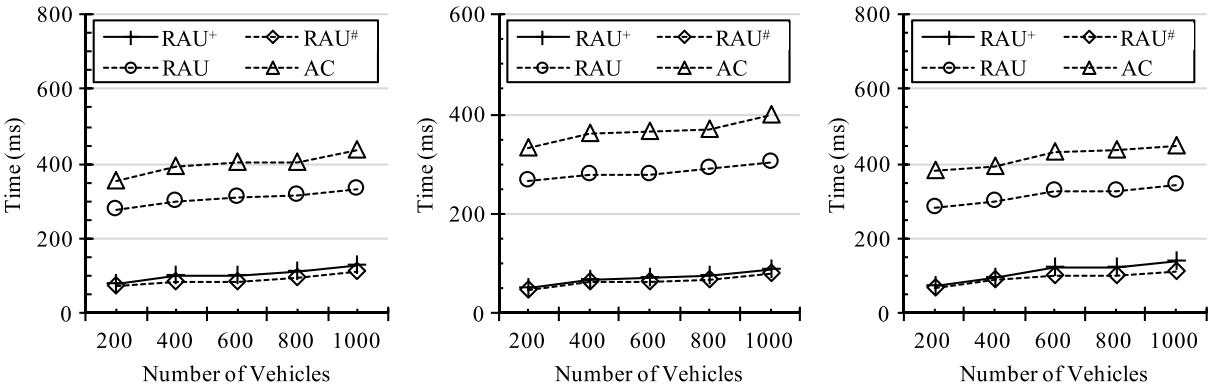
#### 6.2.1. Performance of user registration

The main computational cost involved in the user registration is the generation of the initial pseudonym for the new user. In RAU$^+$ and RAU, each randomized ID is 1024 bits, and for AC system, the length of RSA modulus is 1024 bits. Fig. 6(a) presents the time performance of user registration. We can observe that the RAU is slightly faster than RAU$^+$, and the RAU$^+$ is significantly faster than AC system. This is because the RAU$^+$ needs to perform one more operation, i.e., the generation of the randomized secret key RK associated with the RID, and the AC system needs more exponential computations and much more message exchanges which leads to a longer network delay. Specifically, as for the RAU$^+$ and RAU protocols, without considering network propagation delay, the randomized ID and secret key generation time is less than **23** ms per user. Moreover, besides the standard secure channel establishment, the RAU$^+$ and RAU require 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




(a) Registration time. (b) Total size of packets.

**Fig. 6.** Performance of user registration.




(a) Manhattan. (b) Chicago. (c) Los Angeles.

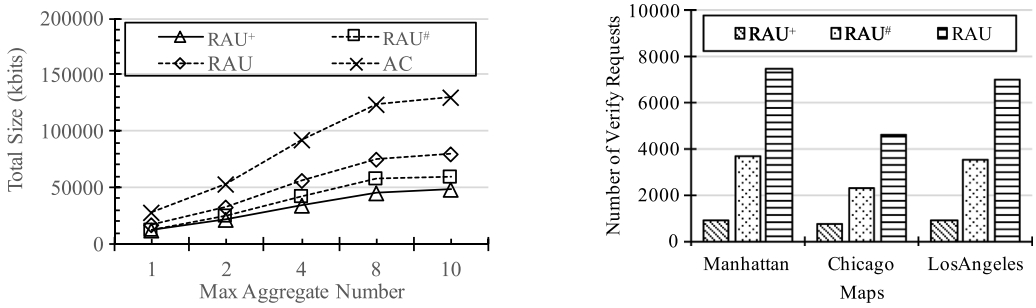
**Fig. 7.** Time performance of user authentication.

and the RS server sends back the initial random ID to the user. However, in the AC system, the pseudonym and credential generation need many time-consuming exponential computations, and 6 rounds or message exchange between the user and the RS server which takes about **380** ms excluding the network delay. Fig. 6(b) shows the total size of transferred packets for registration in network, from which we can see that our RAU$^+$ protocol can significantly reduce the network overhead.

*6.2.2. Performance of user authentication*

Being the best among the exiting solutions, the Anonymous Credential (AC) system [30] achieves most criteria for anonymous authentication as our proposed RAU$^+$ system. However, as discussed in the related work and the security analysis, the AC is not perfectly suitable for VANET applications. One reason is that the AC system assumes that the communication channel is already secure, which is however really challenging for two vehicles before authentication. In this section, we compare the efficiency of (1) the full RAU$^+$ protocol with secure channel establishment, (2) the RAU$^\#$, the part of RAU$^+$ protocol without aggregated identity authentication, (3) the RAU protocol, and (4) the AC system without secure channel establishment.

Fig. 7 reports the running time of each protocol by varying the number of vehicles from 200 to 1000 in three maps. Observe that our proposed RAU$^+$ protocol is significantly faster than the other two approaches. To better understand such behavior, let us review the main steps in each protocol. For each round of authentication, in AC system, the user $i$ shows a single credential to user $j$, and user $j$ shows his/her single credential to user $i$ as well. The computational complexity of the total authentication process is 22 exponentiations as the AC is using the zero-knowledge proof. In our RAU$^+$ protocol, user $i$ and user $j$ need to: (1) exchange their pseudonyms, (2) prove their ownerships of their randomized identity *RID*s, and (3) establish a secure channel between them. It is worth noting that the first phase of our RAU$^+$ protocol already offers the same functionality as the AC protocol. With the additional two phases, RAU$^+$ provides more security guarantee than the AC system while still more than 10 times faster than the AC protocol. Compared to the previous RAU protocol, the RAU$^+$ protocol is also more efficient because the RAU$^+$ integrates the ownership validation in the authentication protocol via the random secret key *RK* and hence reduces the computational and communication time. The RAU$^+$ protocol is slightly slower than RAU$^\#$ because the aggregated identity verification needs to wait a group of users' responses before sending the aggregate




(a) Total data size of authentication.

(b) Total number of verification requests.

**Fig. 8.** Transmission performance of user authentication.

**Table 1**
Identity tracing time.

| Protocol | Real-time tracing | Historical tracing |
|---|---|---|
| RAU$^+$ | 1.8 ms | 3.49 s |

verification data to VS. However, the RAU$^+$ protocol can significantly reduce the overhead of the network and the VS server, which will be discussed below.

Next, we compare the communication complexity in terms of the amount of messages and the total size exchanged throughout the protocol. Fig. 8 shows the number and total size of transmitted packets of user authentication by varying the maps and the number of vehicles requesting simultaneous authentication. As shown in Fig. 8(a), the total number of packets transmitted during the authentication phase (including the communication with the servers) in RAU$^+$ is significantly less than the AC system, the original RAU protocol and the RAU$^\#$ protocol. In our RAU$^+$ protocol, there are total $(1 + 1.5 * k)$ rounds of communication including one round between $u_i$ and the VS server and 1.5 rounds between $u_i$ and each of its $k$ neighboring users $(u_1, u_2, \ldots, u_k)$ for simultaneously two-way authentication. However, in the AC system, there are three rounds of communication between each pair of vehicles and the total is $3k$. Without aggregated identity authentication, the RAU$^\#$ needs $2.5k$ rounds of communications for k pairs of users.

Moreover, the size of the messages in the RAU$^+$ protocol is $(12k + 3)l$ bits ($l$ is the key size) which is also smaller than AC system ($30kl$ bits), the RAU ($22kl$ bits) and the RAU$^\#$ ($15kl$ bits). As expected, in Fig. 8(b), we observe that the advantage of aggregating authentication in RAU$^+$ has become more prominent with the increase of the vehicles requesting for authentication simultaneously. Since part of the communication cost in the RAU and RAU$^+$ involves the server, we take a further look at it in Fig. 8(c). Observe that the aggregated identity authentication in RAU$^+$ has largely reduced the number of authentication requests so that the number of connections established between user and VS can be reduced. All of these is because the aggregated identity authentication (1) avoids sending duplicate messages occurring in a group of pair-wise authentication, (2) reduces the number of connection establishment between users and the VS server (only one establishment is needed), and (3) transfers less bits between users and the VS server ($(6k + 3)l$ bits), which is both smaller than the RAU ($12kl$ bits) and the *RAU*$^\#$ ($9kl$ bits).

*6.2.3. Performance of identity tracing*

Our proposed RAU$^+$ and RAU protocols have a nice feature of providing user tracing in terms of any dispute while the AC system does not. There are two types of tracing available in the RAU$^+$ system: (i) the real-time identity tracing; and (ii) the historical identity tracing. Table 1 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.

## 7. Conclusion

In this paper, we present a highly efficient randomized authentication system in VANETs, namely RAU$^+$. The proposed RAU$^+$ 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. The experiment results demonstrates that our proposed RAU$^+$ protocol is more efficient than other protocols, and can effectively reduce the overhead of the network and the servers' workload. By further adjustment and improvement, our protocol may be extended to IoT schemes or other application scenarios.

## Acknowledgment

This work is partially supported by National Science Foundation under the project DGE-1433659.

## References

[1] M. Gerla, M. Gruteser, Vehicular networks: Applications, protocols, and testbeds, Emerg. Wirel. Technol. Future Mobile Int. (2011) 201–241.
[2] S. Mathur, T. Jin, N. Kasturirangan, J. Chandrasekaran, W. Xue, M. Gruteser, W. Trappe, Parknet: drive-by sensing of road-side parking statistics, in: Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services, ACM, 2010, pp. 123–136.
[3] W. Viriyasitavat, F. Bai, O.K. Tonguz, Toward end-to-end control in vanets, in: Vehicular Networking Conference (VNC), IEEE, 2011, pp. 78–85.
[4] D. Lin, E. Bertino, R. Cheng, S. Prabhakar, Location privacy in moving-object environments, Trans. Data Priv. 2 (1) (2009) 21–46.
[5] M.H. Eiza, Q. Ni, An evolving graph-based reliable routing scheme for vanets, IEEE Trans. Veh. Technol. 62 (4) (2013) 1493–1504.
[6] W. Jiang, F. Li, D. Lin, E. Bertino, No one can track you: Randomized authentication in vehicular ad-hoc networks, in: 2017 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, 2017, pp. 197–206.
[7] R. Lu, X. Lin, H. Zhu, P.-H. Ho, X. Shen, Ecpp:efficient conditional privacy preservation protocol for secure vehicular communications, in: Proc. of IEEE Conference on Computer Communications, 2008, pp. 1229–1237.
[8] A. Squicciarini, D. Lin, A. Mancarella, Paim: Peer-based automobile identity management in vehicular ad-hoc network, in: Computer Software and Applications Conference (COMPSAC), 2011 IEEE 35th Annual, IEEE, 2011, pp. 263–272.
[9] C. Zhang, R. Lu, X. Lin, P. Ho, X. Shen, An efficient identity-based batch verification scheme for vehicular sensor networks, in: INFOCOM 2008. the 27th Conference on Computer Communications, IEEE, 2008, pp. 246–250.
[10] M. Raya, J.-P. Hubaux, Securing vehicular ad hoc networks, J. Comput. Secur. 15 (1) (2007) 39–68.
[11] 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.
[12] M. Raya, D. Jungels, P. Papadimitratos, I. Aad, J.-P. Hubaux, Certificate revocation in vehicular networks Laboratory for Computer Communications and Applications (LCA) School of Computer and Communication Sciences, EPFL, Switzerland, 2006.
[13] G. Calandriello, P. Papadimitratos, J.-P. Hubaux, 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, J. Su, An efficient pseudonymous authentication scheme with strong privacy preservation for vehicular communications, IEEE Trans. Veh. Technol. 59 (7) (2010) 3589–3603.
[15] A. Studer, E. Shi, F. Bai, 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, in: SECON'09, IEEE Press, Piscataway, NJ, USA, 2009, pp. 484–492.
[16] J. Sun, C. Zhang, Y. Zhang, Y. Fang, An identity-based security system for user privacy in vehicular ad hoc networks, IEEE Trans. Parallel Distrib. Syst. 21(9) (2010) 1227–1239.
[17] C. Zhang, P.-H. Ho, J. Tapolcai, On batch verification with group testing for vehicular communications, Wirel. Netw. 17(8) (2011) 1851–1865.
[18] J. Zhang, Y. Cui, Z. Chen, SPA: Self-certified pkc-based privacy-preserving authentication protocol for vehicular ad hoc networks, Int. J. Secur. Appl. 6 (2) (2012) 409–414.
[19] J. Li, H. Lu, M. Guizani, Acpn: a novel authentication framework with conditional privacy-preservation and non-repudiation for vanets, IEEE Trans. Parallel Distrib. Syst. 26 (4) (2015) 938–948.
[20] K. Sha, Y. Xi, W. Shi, L. Schwiebert, T. Zhang, Adaptive privacy-preserving authentication in vehicular networks, in: ChinaCom'06. First International Conference on Communications and Networking in China, 2006, IEEE, 2006, pp. 1–8.
[21] C.D. Jung, C. Sur, Y. Park, K.-H. Rhee, A robust conditional privacy-preserving authentication protocol in vanet, Soc. Inform. Telecommun. Eng. 17 (2009) 35–45.
[22] Y. Wang, H. Zhong, Y. Xu, J. Cui, Ecpb: Efficient conditional privacy-preserving authentication scheme supporting batch verification for vanets, Int. J. Netw. Secur. 18 (2) (2016) 374–382.
[23] Y. Hao, C. Yu, C. Zhou, W. Song, A distributed key management framework with cooperative message authentication in vanets, IEEE J. Sel. Areas Commun. 29 (3) (2011) 616–629.
[24] F. Wang, Y. Xu, H. Zhang, Y. Zhang, L. Zhu, 2flip: A two-factor lightweight privacy-preserving authentication scheme for vanet, IEEE Trans. Veh. Technol. 65 (2) (2016) 896–911.
[25] L. Yeh, Y. Chen, J. Huang, Paacp: A portable privacy-preserving authentication and access control protocol in vehicular ad hoc networks, Comput. Commun. 34 (3) (2011) 447–456.
[26] Z. Tan, A lightweight conditional privacy-preserving authentication and access control scheme for pervasive computing environments, J. Netw. Comput. Appl. 35 (6) (2012) 1839–1846.
[27] C. Gamage, B. Gras, B. Crispo, A. Tanenbaum, An identity-based ring signature scheme with enhanced privacy, in: Securecomm and Workshops, 2006, pp. 1–5.
[28] S. Zeng, Y. Huang, X. Liu, Privacy-preserving communication for vanets with conditionally anonymous ring signature, Int. J. Netw. Secur. 17 (2) (2015) 135–141.
[29] U. Rajput, F. Abbas, J. Wang, H. Eun, 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.
[30] J. Camenisch, E.V. Herreweghen, Design and implementation of the *idemix* anonymous credential system, in: ACM Conference on Computer and Communications Security, 2002, pp. 21–30.
[31] J. Camenisch, A. Lysyanskaya, An efficient system for non-transferable anonymous credentials with optional anonymity revocation, in: EUROCRYPT, 2001, pp. 93–118.
[32] J. Camenisch, 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.
[33] J. Camenisch, T. Groß, Efficient attributes for anonymous credentials, in: Proceedings of the 15th ACM Conference on Computer and Communications Security, ACM, 2008, pp. 345–356.
[34] E. Brickell, J. Camenisch, L. Chen, Direct anonymous attestation, in: Proceedings of the 11th ACM Conference on Computer and Communications Security, ACM, 2004, pp. 132–145.
[35] E. Cesena, H. Löhr, G. Ramunno, A.-R. Sadeghi, D. Vernizzi, Anonymous authentication with tls and daa, in: Trust and Trustworthy Computing, 2010, pp. 47–62.
[36] C. Wachsmann, L. Chen, K. Dietrich, H. Löhr, A.-R. Sadeghi, J. Winter, Lightweight anonymous authentication with tls and daa for embedded mobile devices, in: Information Security, 2011, pp. 84–98.

[37] J. Camenisch, A. Lysyanskaya, Dynamic accumulators and application to efficient revocation of anonymous credentials, in: Advances in Cryptology - CRYPTO 2002, Springer, 2002, pp. 61–76.
[38] S. Goldwasser, S. Micali, 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.
[39] P. Paillier, Public key cryptosystems based on composite degree residuosity classes, in: Advances in Cryptology - Eurocrypt '99 Proceedings, LNCS 1592, Springer-Verlag, Prague, Czech Republic, 1999, pp. 223–238.