A Traitor-resistant and Dynamic Anonymous Communication Service for Cloud-based VANETs

Huiying Hou, Jianting Ning, Yunlei Zhao, Robert H. Deng, *Fellow, IEEE,*

Abstract—Cloud-based VANETs are designed to enable communication between high-speed vehicles. In such a highly dynamic environment, how to provide secure and anonymous communication service is a challenge. In this paper, we affirmatively address the challenge by proposing a traitor-resistant and dynamic anonymous communication framework (TD-ACF) for cloud-based VANETs, which supports several advantageous features. In TD-ACF, each vehicle is represented by a set of attributes instead of its real identity, and the driving data is transmitted in encrypted form. Therefore, the anonymous authentication and the confidentiality of driving data are achieved in this way. Meanwhile, TD-ACF supports two practical requirements in cloud-based VANETs: the revocation and the traceability of traitor. For the former, TD-ACF allows a vehicle to exit the communication network at any moment. We employ an efficient binary tree algorithm to reduce the size of key updates for revocation from the traditional linear to the logarithmic level. For the latter, we overcome the barrier of the one-to-many relationship between a vehicle and the shared set of attributes to support traitor tracing. In TD-ACF, unlike most existing schemes, the Semi-Trusted Cloud (STC) can directly capture and punish a traitor instead of querying all the records in the list of unrevoked vehicles. In addition, we solve the key escrow problem that plagues most existing attribute-based schemes. The theoretical analysis and experimental simulation show that the proposed scheme is feasible and effective.

Index Terms—Cloud Services, Cloud-based VANETs, Dynamic, Attributed-based, Traceability, Key Escrow Resistance.

1 Introduction

The wide application of cloud services has promoted the development of many industries and brought great convenience to people's life. As one of the cloud services, cloud-based VANET has attracted great attention from both industry and academia, which is the most promising technologies for ensuring traffic efficiency and security. Cloud-based VANET is a huge interactive network carrying critical traffic information such as vehicles' locations, speed, and routes. A vehicle can wirelessly send traffic information to nearby vehicles or Road-Side Units (RSUs) in cloud-based VANET. In this way, each nearby vehicle can better understand the driving environment and adjust the driving plan as needed. However, these advantages are not free, and the widespread use of cloud-based VANET raises concerns about the privacy of vehicle's identity and driving data.

In order to protect the privacy of vehicle's identity, many anonymous authentication schemes [1–5] based on pseudonym have been proposed. However, under the pseudonym mechanism, each vehicle needs to store a large number of pseudonyms, which causes heavy storage burden. In order to reduce such storage overhead, several schemes based on group or anonymous certificate [6, 7] have been proposed. As VANET is highly dynamic, combining high-speed vehicles into a group is a great challenge, and there exists no satisfactory solution so far. The anonymous certificate-based scheme suffers from the problem of certificate abuse, where malicious vehicles can simultaneously use a set of different anonymous certificates to simulate multiple vehicles.

In practice, a vehicle may exit the network due to expiration of contract period or the software/hardware malfunctions. Malicious vehicles may broadcast wrong traffic information to mislead surrounding vehicles with the intention to cause harm or for financial gains. None of the above mentioned schemes support dynamic revocation of vehicle and traceability of traitor.

To realize secure and dynamic anonymous communication for VANET, *Cui et al*. [8] proposed an attribute-based framework (RT-ABS for short), which supports vehicle revocation and the traceability of traitor. In RT-ABS, each vehicle is represented by a set of attributes that are shared among multiple vehicles. Therefore, vehicle's real identity can be hidden through the one-to-many relationship between a set of attributes and vehicles. To support the revocation of vehicle, RT-ABS incorporates a binary tree structure into the key generation and key update processes, which reduces the size of key updates from linear to logarithmic. The resulting long-term attribute-key size is large, which incurs a heavy storage burden to the resource-constrained vehicle. To support the traceability of traitor, in the scheme [8], the introduced Trusted Authority (TA) needs to store a list of

- *H. Y. Hou is with the College of Computer Science and Technology, Fudan University, Shanghai, China, 200000. E-mail:18110240036@fudan.edu.cn*
- *J. T. Ning is with the College of Mathematics and Informatics, Fujian Normal University, Fuzhou, China, 350117. E-mail:jtning88@gmail.com.*
- *Y. L. Zhao is with the College of Computer Science and Technology, Fudan University Shanghai, China, 200000. E-mail:ylzhao@fudan.edu.cn.* (Corresponding author)
- *R. H. Deng is with the Secure Mobile Centre, School of Information Systems, Singapore Management University, Singapore, 178902. E-mail: robertdeng@smu.edu.sg*

Manuscript received June 6, 2020; revised February 07, 2021.

unrevoked vehicles, and to query each record in the list to capture traitor. As a result, the efficiency of traceability is low. Another weakness of RT-ABS is that TA can impersonate any vehicle to generate a valid signature under certain conditions. In addition, RT-ABS (and actually all the above schemes) does not take into account the confidentiality of driving data. The leakage of driving data will directly expose the personal privacy of drivers [9].

To simultaneously support anonymous authentication and confidentiality of driving data, several schemes have been proposed [10–12]. The scheme in [10] integrates pseudonym with Identity-Based Signature (IBS) to support anonymous authentication, and adopts ciphertext policy attribute-based encryption (CP-ABE) to provide access control of driving data. However, this scheme causes heavy storage burden on the vehicle. The scheme in [11] presents a pre-authentication method for secure communications in cloud-based VANETs. This scheme employs symmetric encryption to realize the confidentiality of driving data. However, symmetric encryption is in general not suitable for the communication in cloud-based VANETs as vehicles have to contact a base station to decrypt/verify messages received from another vehicle. The scheme in [12] is not practical since combining high-speed vehicles into a group is very difficult in reality. Neither of the above-mentioned schemes takes into account the two practical requirements considered in this work, namely, the revocation of vehicle and the traceability of traitor. Therefore, how to design a practical scheme for cloud-based VANETs, which simultaneously supports anonymous authentication and confidentiality of driving data, still needs to be further explored.

Our contributions. In this paper, we investigate the problem of providing a secure and practical anonymous communication service for cloud-based VANETs, and propose a traitor-resistant and dynamic anonymous communication framework (TD-ACF) for cloud-based VANETs, which supports anonymous authentication and confidentiality of driving data simultaneously. To support highly efficient revocation of vehicle and the traceability of traitor, TD-ACF provides efficient solutions whereby (a) employing an efficient binary tree algorithm for revocation so that the key size that stored on the resource-constrained vehicle is dramatically reduced compared to the scheme [8], and (b) introducing a hidden identity such that the traitor can be captured by STC directly. In particular, the contribution of this paper can be summarized as follows:

- We propose a traitor-resistant and dynamic anonymous communication framework based on attribute-based signcryption (ABSC) to provide a secure and practical anonymous communication service for cloud-based VANETs. In TD-ACF, each vehicle is represented by a set of attributes that are shared by multiple vehicles, and the driving data is transmitted in encrypted form. In this way, the anonymous authentication and the confidentiality of driving data can be supported simultaneously in TD-ACF. TD-ACF also guarantees that no one, including malicious adversaries and STC, can impersonate a vehicle to communicate with others. In addition, TD-ACF supports two practical requirements in cloud-based VANETs: the revocation and the traceability of traitor. In particular, in TD-ACF, STC can precisely capture and punish traitor. More pragmatically, TD-ACF may remove a vehicle from the VANET at any moment.
- We design a traceable and revocable attribute-based signcryption scheme. In order to realize efficient revocation, we employ an efficient binary tree algorithm to reduce the size of key updates from the traditional linear to the logarithmic level without expanding the key size. In order to support traceability, we insert the hidden identity into our ABSC scheme such that STC can directly capture traitor from the hidden identity information instead of querying each record in the unrevoked vehicle list as the conventional solutions do. In addition, we solve the key escrow problem that perplexes all existing attribute-based signcryption schemes.
- We give the security model and definitions, and prove the security of the proposed scheme. Then, we evaluate its performance through extensive experiments. Experimental results show that the proposed scheme is suitable for practical applications.

Organization. The rest of this paper is organized as follows. In Section 2, we introduce the work related to the secure communication in cloud-based VANETs. In Section 3, we briefly review the preliminary knowledge related to this paper. We give the definition of system model and security model in Section 4. In Section 5, we describe the proposed scheme in detail. In Section 6, we introduce the formal proof of the security of the proposed scheme. We evaluate the performance of the proposed scheme in Section 7. Finally, we come to the conclusion in Section 8.

2 Related work

In order to avoid the exposure of vehicle's driving path, many privacy-preserving schemes [10, 13–28] that are based on public key cryptography, identity-based cryptography, group signature, anonymous certificate, and attribute-based encryption, have been proposed for VANET. However, all of the aforementioned schemes incur heavy computation and storage burden from the frequent updates of pseudonyms or private keys.

In public key-based schemes [14], [18] and [29], a Public Key Infrastructure (PKI) is required to issue public key certificates to vehicles. Due to the frequent change of pseudonyms, in such schemes, each vehicle has to store a large number of public-private key pairs and public key certificates, which incur heavy storage burden. Compared with public key based schemes, public key certificates are no longer required in identity based schemes [19], [30] and [31]. The identity-based schemes rely on a trusted authority to generate pseudonym, and the vehicle also needs to store large numbers of pseudonym in advance. In order to reduce the storage burden of vehicle, a series of schemes [7], [20], [21], [27], [28], [6] and [32] based on group signature and anonymous certificate have been proposed. All aforementioned schemes are based on bilinear pairing. The scheme in [33] points out that attribute-based schemes based on lattices

and based on multilinear maps are not at all efficient yet. For practical considerations, attribute-based schemes [34, 35] in bilinear group setting supporting monotone access policies (i.e., monotone span programs) currently seem to be the optimal choice. However, a common disadvantage of such schemes is that when a vehicle is revoked, the size of the key update is linear to the number of unrevoked vehicles, and a revocation list is needed to check whether the vehicle has been revoked or does not.

In order to support efficient revocation for VANET, *Cui et al*. [8] presents a dynamic attribute-based communication framework based on attribute-based signature. However, this scheme does not take the confidentiality of driving data into account [9], [10]. In order to simultaneously support anonymous authentication and confidentiality of driving data, several schemes have been proposed [10-12]. However, these schemes do not take into account two practical requirements, namely, the revocation of vehicle and the traceability of traitor.

To the best of our knowledge, all of the existing communication schemes for VANET cannot support anonymous authentication, confidentiality of driving data, traceability and revocability simultaneously. In this paper, we explore how to design a secure and practical anonymous communication scheme for cloud-based VANETs, which supports anonymous authentication, confidentiality of driving data, traceability and revocability simultaneously.

3 PRELIMINARIES

3.1 Notions

We show some notations used in the description of our scheme in Table 1.

TABLE 1: **Notations**

Notation	Meaning
p	One large prime
G, G_T	Multiplicative cyclic groups with the prime order p
g	A generator of group G
$\hat{e}$	A bilinear map $\hat{\mathrm{e}} : G \times G \rightarrow G_T$
Z_p^*	A prime field with nonzero elements
λ	The security parameter
msk	The master private key
mpk	The master public key
A	The attributes set
VID	The vehicle's identity
pk_{vid}	The public user-key
sk_{vid}	The private user-key
$sk_{vid,A}$	The attribute-key
G_i	The set of unrevoked vehicles
$keyup$	The key update information
$PathKey$	The path key
β	The private user-key
m	The message
t	The timestamp

3.2 Bilinear Groups

Let $\Phi(1^\lambda)$ be an algorithm, which takes security parameter λ as input and outputs a symmetric bilinear map of the prime order p. Let $(p, G, G_{\mathrm{T}}, \hat{\mathrm{e}})$ denote the output of the algorithm $\Phi(1^\lambda)$, where G and G_{T} are two multiplicative cyclic groups with the prime order p. And the bilinear map $\hat{\mathrm{e}} : G \times G \rightarrow G_T$ satisfies the following characteristics:

- ***Bilinearity***: $\hat{\mathrm{e}}(u^a, v^b) = \hat{\mathrm{e}}(u^b, v^a) = \hat{\mathrm{e}}(u, v)^{ab}$ for $\forall u, v \in G$ and $\forall a, b \in \mathrm{Z}_p$.
- ***Non-degeneracy***: $\hat{\mathrm{e}}(g, g) \neq 1$, g is the generator of the group G.

If operations in group G and bilinear map $\hat{\mathrm{e}} : G \times G \rightarrow G_T$ can be efficiently computed, then the group G is said to be a bilinear group. We note that, for simplicity, we construct our scheme on symmetric bilinear groups, but it can be extended to asymmetric bilinear groups in a standard way.

3.3 Hardness Assumption

The proposed scheme is based on Decisional Bilinear Diffie-Hellman (DBDH) assumption, Computational Diffie-Hellman (CDH) assumption and Discrete Logarithm (DL) assumption in bilinear group. Below, we briefly review these two assumptions.

Definition 1 (DBDH Assumption). Assume $\hat{\mathrm{e}} : G \times G \rightarrow G_T$ is a bilinear map and g be a generator of the group G. Then, u, v, w are three random numbers in Z_p, and h is a random number in the group $\mathrm{G_T}$. The DBDH assumption holds if no polynomial time adversary can distinguish the two tuples $(A = g^u, B = g^v, C = g^w, Z = \hat{e}(g, g)^{uvw})$ and $(A = g^u, B = g^v, C = g^w, Z = \hat{e}(g, g)^{uvw})$ with non-negligible probability.

Definition 2 (CDH Assumption). Assume G be a group of prime order p. Let g be a generator of the group G. The CDH assumption is that, given a tuple (g, g^u, g^v), where $u, v \in \mathrm{Z}_p$ are secret, there exist no polynomial time algorithm to compute g^{uv} with non-negligible probability.

Definition 3 (DL Assumption). Assume G be a group of prime order p. Let g be a generator of the group G. The DL assumption is that, given a pair (g, g^x), where $x \in \mathrm{Z}_p^*$, there exist no polynomial time algorithm to compute x with non-negligible probability.

4 PROBLEM FORMULATION

4.1 System Model

In order to provide a secure and practical anonymous communication service for cloud-based VANETs, we proposed a traitor-resistant and dynamic anonymous communication framework (TD-ACF). The system model consists of three kinds of different entities: the Semi-Trusted Cloud (STC), the On-Board Units (OBUs), and the Road Side Units (RSUs), as shown in Fig.1.

1) STC: The STC is honest-but-curious and has rich storage and computational resources. STC is responsible for generating attribute key for vehicle according to its set of possessed attributes, and issuing messages of key update periodically. In addition, STC plays the role of traffic control center and can trace the traitor.

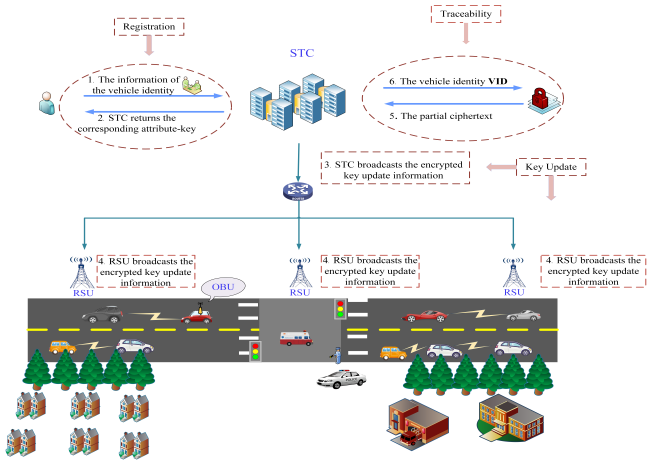

Fig. 1: System model

2) OBUs: The OBUs are wireless devices. The vehicle wirelessly communicates with other vehicles and RSUs through the equipped OBUs.
3) RSUs: The RSUs collect messages sent by vehicles in its jurisdiction, and further send these messages to the traffic control center for traffic control. In this paper, STC also plays the role of traffic control center. The communication between RSUs and STC is carried out through wired network.

In the system framework, the revocable and traceable attribute-based signcryption scheme proposed in this paper is used to realize secure communication between vehicles, and vehicle and RSUs. Each vehicle is described as a set of attributes (for example, District: Yangpu, City: Shanghai, Country: China, Expiry: 062020, etc.). The set contains no information about the vehicle's identity **VID**. In order to specify the system in more detail, we introduce the specific workflow of the system below:

- First, when a vehicle wants to join cloud-based VANET, it can register with STC offline. This operation only needs to be done once, and the offline mode fully prevents the private information of the vehicle from being known to anyone except STC. The vehicle first submits the public user-key, vehicle's identity **VID** and the set of owned attributes to STC. STC then stores the vehicle's identity **VID** and generates a long-term attribute-key for the vehicle based on its set of possessed attributes.
- Then, STC periodically generates key update messages and enforces this operation when a vehicle exits the system. When STC discovers that a vehicle is misbehaving or corrupted, it forcibly removes it from the system. STC broadcasts the encrypted key update message to each RSU, and each RSU broadcasts the ciphertext to all vehicles in its jurisdiction. However, only unrevoked vehicles can decrypt the key update message correctly.
- After receiving the ciphertext, the unrevoked vehicles can correctly decrypt the key update message and calculate the new attribute-key. The revoked vehicles cannot obtain the new attribute-key of the current time period. Thus, the revoked vehicles cannot generate a valid ciphertext.
- The vehicle uses its own private user-key and the attribute-key of the current time period to generate ciphertext. Since the private user-key is only known by the vehicle itself, even STC with the master private key cannot impersonate it to generate a valid ciphertext.
- Finally, STC with the master private key can directly trace back to sender's identity **VID** according to the hidden identity contained in the ciphertext.

4.2 Design Goals

To efficiently support anonymous authentication, confidentiality of driving data, traceability and revocability simultaneously for communication in cloud-based VANETs, our scheme is designed to achieve the following goals:

- ***Confidentiality of Driving Data:*** to ensure that the driving data is not exposed to malicious adversary.
- ***Unforgeability:*** to ensure that no one, including malicious adversaries and STC, can impersonate a vehicle to generate a valid signcryption ciphertext.
- ***Anonymity:*** to ensure that anyone, except STC, can know no information about the sender's identity according to the received ciphertext, and cannot link multiple ciphertexts to the vehicle that sent them.
- ***Traceability.*** to assure that STC with the master private key can directly trace back to the vehicle according to the given ciphertext.

4.3 Definitions

4.3.1 Scheme Definition

A traitor-resistant and dynamic anonymous communication scheme for cloud-based VANETs consists of the following eight algorithms: ***Setup***, ***KeyGen***, ***KeyUpdate***, ***Signcryptkey***, ***Signcrypt***, ***Unsigncrypt***, ***Revoke*** and ***Trace***. The above algorithms are described as follows in detail:

- ***Setup*** (1^{λ}): This algorithm takes the security parameter λ as input, and outputs the public parameter *mpk* and the master private key *msk*.

- ***KeyGen*** (mpk, msk, **A**)***:*** This algorithm takes the master private key msk, the attributes set **A** and the public parameter mpk as input, and outputs the attribute-key $sk_{vid,\mathrm{A}}$, the public user-key pk_{vid} and the private user-key sk_{vid}.
- ***KeyUpdate*** (msk, G_i)***:*** This algorithm takes the master private key msk, the set G_i of unrevoked vehicles as input, and outputs the encrypted key update information $keyup$.
- ***SigncryKG*** ($sk_{vid,\mathrm{A}}$, $keyup$, $PathKey$)***:*** This algorithm takes the attribute-key $sk_{vid,\mathrm{A}}$, the key update information $keyup$ and the path key $PathKey$ as input, and outputs the new attribute key $newkey$.
- ***Signcrypt*** (w_e,$\Gamma_{k,S}$, $sk_{vid,A}$, $pk_{vid,A}$, $newkey$, β, m, t, **VID**)*:* This algorithm takes the attributes set w_e, the claim predicate $\Gamma_{k,S}$, the attribute-key $sk_{vid,\mathrm{A}}$, the new attribute key $newkey$, the private user-key $\beta \leftarrow Z_p^*$, the message m, the timestamp t and the vehicle's identity **VID** as input, and outputs the ciphertext $\boldsymbol{C}$.
- ***Unsigncrypt*** (w_e,$\Gamma_{k,S}$, sk_{vid',A_r}, mpk, β', t, $\boldsymbol{C}$)*:* This algorithm takes the public parameter mpk, the attributes set w_e, the claim predicate $\Gamma_{k,S}$, the receiver's attribute-key sk_{vid',A_r}, the private receiver's user-key β', and the timestamp t as input, and outputs the message m.
- ***Revoke*** (t, ***VID***)***:*** This algorithm takes the timestamp t and the vehicle's identity **VID** as input, and outputs the updated set G_i of unrevoked vehicles.
- ***Trace*** (t, AID)***:*** This algorithm takes the timestamp t and the hidden identity AID as input, and outputs sender's identity **VID**.

4.3.2 Security Definitions

The definitions of security is divided into the following four categories:

Definition 4 (Confidentiality of Driving Data). Confidentiality ensures that the driving data from cannot be eavesdropped during the communication process. In order to formally describe the confidentiality of driving data, we introduce a game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$, which tries to obtain the driving data with unauthorized attributes set. STC is viewed as a challenger $\mathcal{C}$ and the malicious receiver is viewed as an adversary $\mathcal{A}$ in our security definition. This game includes the following phases:

- **Setup Phase:** Firstly, the adversary $\mathcal{A}$ selects the specified attributes set w_e^* and submits it to the challenger $\mathcal{C}$. Then, the challenger $\mathcal{C}$ performs ***Setup*** algorithm, and sends the common parameters mpk to $\mathcal{A}$.
- **Query Phase:** The adversary $\mathcal{A}$ makes following six queries to the challenger $\mathcal{C}$.

  1) *KeyGen Queries*: The adversary $\mathcal{A}$ makes queries the attribute key for the attributes set w_r, where $|w_r \cap w_e^*| < d$ and d is an integer representing the threshold of unsigncryption. The challenger $\mathcal{C}$ runs ***KeyGen*** algorithm and returns attribute-key to $\mathcal{A}$.

  2) *KeyUpdate Queries*: The adversary $\mathcal{A}$ queries key update information for the time period tp. Then, the challenger $\mathcal{C}$ performs ***KeyUpdate*** algorithm and sends the key update information $keyup$ to $\mathcal{A}$.

  3) *SigncryKG Queries*: The adversary $\mathcal{A}$ queries the signcryption key for the vehicle's identity **VID**, the attributes set w_A and the time period tp. Then, the challenger $\mathcal{C}$ performs ***SigncryKG*** algorithm and sends the signcryption key to $\mathcal{A}$.

  4) *Signcrypt Queries*: The adversary $\mathcal{A}$ queries the ciphertext for the the attributes set w_e, the claim predicate $\Gamma_{k,S}$, the attribute-key sk_{vid,w_A}, the new attribute key $newkey$, the private user-key β, the message m, the timestamp t and the vehicle's identity **VID**. Then, the challenger $\mathcal{C}$ performs ***Signcrypt*** algorithm and sends the ciphertext $\boldsymbol{C}$ to $\mathcal{A}$.

  5) *Unsigncrypt Queries*: The adversary $\mathcal{A}$ queries the message for the public parameter mpk, the attributes set w_e, the claim predicate $\Gamma_{k,S}$, the attribute-key sk_{vid',w_r}, the ciphertext $\boldsymbol{C}$, the private user-key β', and the timestamp t. Then, the challenger $\mathcal{C}$ performs ***Unsigncrypt*** algorithm and sends the message m to $\mathcal{A}$.

  6) *Revoke Queries*: The adversary $\mathcal{A}$ queries the revocation list for the vehicle **VID**. The challenger $\mathcal{C}$ performs ***Revoke*** algorithm to remove **VID** from the revocation list G_i and returns it to $\mathcal{A}$.
- **Challenge Phase:** In this phase, the adversary $\mathcal{A}$ selects two messages m_0 and m_1 of the same length. The challenger $\mathcal{C}$ randomly selects $b \leftarrow \{0,1\}^*$, and generates the challenge ciphertext $C^* = Signcrypt(m_b, w_e^*, \Gamma_{k,S}, sk_{vid,w_\mathrm{A}}, pk_{vid,w_\mathrm{A}}, \beta, t, \mathbf{VID})$. Finally, the challenger $\mathcal{C}$ returns the challenge ciphertext C^* to $\mathcal{A}$.
- **Guess Phase:** Firstly, the adversary $\mathcal{A}$ performs polynomial queries as in **Query Phase**. But when $|w_e^* \cap w_r| \geq d$, the adversary $\mathcal{A}$ is not allowed to make *keyGen* queries and *Unsigncrypt* queries on the attributes set w_r. Finally, the adversary $\mathcal{A}$ returns a bit b'.

In the above game, it formalizes that if the adversary $\mathcal{A}$, who owns the set of attributes does not meet the access control structure of a ciphertext, should not decrypt the ciphertext correctly. The adversary's goal is to correctly guess the encrypted message m_b in the challenge ciphertext C^* generated by the challenger. The adversary $\mathcal{A}$ can obtain the driving data with the probability $\Pr[b' = b] - \frac{1}{2}$ in this game. We say that the proposed scheme realizes the confidentiality of driving data if for any polynomial time adversary $\mathcal{A}$, $\left|\Pr\left[b' = b\right] - \frac{1}{2}\right| < 1/poly(n)$ for a sufficiently large n, where $poly$ stands for a polynomial function.

Definition 5 (Unforgeability). Unforgeability ensures that unauthorized signers cannot forge a valid signature. In order to formally describe the unforgeability, we introduce a game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, which does not own the designated attributes set. Here, we define two types of adversaries $\mathcal{A}_1$ and $\mathcal{A}_2$. The adversary $\mathcal{A}_1$ does not know the master private key, and the $\mathcal{A}_2$ is STC which holds the master private key.

Firstly, we describe the game played with $\mathcal{A}_1$ and the challenger $\mathcal{C}$. The game consists of the following three phases:

- **Setup Phase:** Firstly, the adversary $\mathcal{A}$ selects the specified attributes set $\{S' \cap \Omega'\}^*$ and submits it to challenger $\mathcal{C}$. Then, the challenger $\mathcal{C}$ performs the ***Setup*** algorithm, and sends the common parameters mpk to $\mathcal{A}_1$.
- **Query Phase:** The adversary $\mathcal{A}_1$ makes following six queries to the challenger $\mathcal{C}$.

  1) *KeyGen Queries*: The adversary $\mathcal{A}_1$ makes queries the attribute-key for the attributes set w_r, where $|w_r \cap w_e^*| < d$. The challenger $\mathcal{C}$ runs ***KeyGen*** algorithm and returns attribute-key to $\mathcal{A}_1$.

  2) *KeyUpdate Queries*: The adversary $\mathcal{A}_1$ queries key update information for the time period tp. Then, the challenger $\mathcal{C}$ performs ***KeyUpdate*** algorithm and sends the key update information *keyup* to $\mathcal{A}_1$.

  3) *SigncryKG Queries*: The adversary $\mathcal{A}_1$ queries the signcryption key for the vehicle's identity **VID**, the attributes set w_{A} and the time period tp. Then, the challenger $\mathcal{C}$ performs ***SigncryKG*** algorithm and sends the signcryption key to $\mathcal{A}_1$.

  4) *Signcrypt Queries*: The adversary $\mathcal{A}_1$ queries the ciphertext for the the attributes set w_e, the claim predicate $\Gamma_{k,S}$, the attribute-key $sk_{vid,\mathrm{A}}$, the new attribute key *newkey*, the private user-key β, the message m, the timestamp t and the vehicle's identity **VID**. Then, the challenger $\mathcal{C}$ performs ***Signcrypt*** algorithm and sends the ciphertext to $\mathcal{A}_1$.

  5) *Unsigncrypt Queries*: The adversary $\mathcal{A}_1$ queries the message for the public parameter *mpk*, the attributes set w_e, the claim predicate $\Gamma_{k,S}$, the attribute-key sk_{vid',w_r}, the ciphertext $\boldsymbol{C}$, the private user-key β', and the timestamp t. Then, the challenger $\mathcal{C}$ performs ***Unsigncrypt*** algorithm and sends the message m to $\mathcal{A}_1$.

  6) *Revoke Queries*: The adversary $\mathcal{A}_1$ queries the revocation list for the vehicle **VID**. The challenger $\mathcal{C}$ performs ***Revoke*** algorithm to remove **VID** from the revocation list G_i and returns it to $\mathcal{A}_1$.
- **Forgery Phase:** Finally, the adversary $\mathcal{A}_1$ outputs a forged ciphertext C^* and the attributes set w_e^*.

In the above game, we need to prove that if the adversary $\mathcal{A}_1$, who does not own the set of attributes $\{S' \cap \Omega'\}^*$, can not forge the ciphertext C^*. The ***Unsigncrypt*** algorithm outputs m^*, where $m^* \neq m$, as a result of taking the ciphertext C^* as input. An adversary $\mathcal{A}_1$ can forge a valid signature with probability $\mathrm{Adv}(\mathcal{A}_1) = \Pr[win]$ in this game. We say that the proposed scheme realizes unforgeability if for any polynomial time adversary $\mathcal{A}_1$, the advantage $\mathrm{Adv}(\mathcal{A}_1)$ is negligible.

Next, we describe the game played with the adversary $\mathcal{A}_2$ and the challenger. The game consists of the following three phases.

- **Setup Phase:** The challenger $\mathcal{C}$ performs the ***Setup*** algorithm, and sends the common parameters mpk and the master private key msk to $\mathcal{A}_2$.
- **Private User-Key Queries:** The the adversary $\mathcal{A}_2$ queries the private user-key except for the vehicle **VID**. Then the challenger $\mathcal{C}$ randomly selects $\beta \leftarrow \mathrm{Z}_p^*$, and returns it to $\mathcal{A}_2$.
- **Forgery Phase:** The adversary $\mathcal{A}_2$ generates a forged ciphertext C^* on the time period tp^*, the attributes set w_e^*, the claim predicate $\Gamma_{k,S}^*$ and the message m^*.

In the above game, we need to prove that if the adversary $\mathcal{A}_2$, who does not own the set of attributes $\{S' \cap \Omega'\}^*$, can not forge the ciphertext C^*. The ***Unsigncrypt*** algorithm outputs m^*, where $m^* \neq m$, as a result of taking the ciphertext C^* as input. And the output of the ***Trace*** algorithm is $\mathbf{VID}^*$, where $\mathbf{VID}^* \neq \mathbf{VID}$. An adversary $\mathcal{A}_2$ can forge a valid signature with probability $\mathrm{Adv}(\mathcal{A}_2) = \Pr[win]$ in this game. We say that the proposed scheme realizes unforgeability if for any polynomial time adversary $\mathcal{A}_2$, the advantage $\mathrm{Adv}(\mathcal{A}_2)$ is negligible.

Definition 6 (Anonymity). We say that a communication scheme for cloud-based VANETs achieves anonymity if anyone including valid receivers cannot derive any identity information of the sender. In order to formally describe the anonymity, we introduce a game between the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$, who does not know the master private key. STC is viewed as a challenger $\mathcal{C}$. This game includes the following phases:

- **Setup Phase:** Firstly, the adversary $\mathcal{A}$ selects the specified attributes set w_e^* and submits it to the challenger $\mathcal{C}$. Then, the challenger $\mathcal{C}$ performs ***Setup*** algorithm, and sends the common parameters mpk to $\mathcal{A}$.
- **Query Phase:** The adversary $\mathcal{A}$ performs as in **Query Phase** of the Definition 4.
- **Output Phase:** The adversary $\mathcal{A}$ outputs the vehicle's identity VID^*.

The above game formalizes that if the adversary $\mathcal{A}$, who does not know the master private key, should not obtain the vehicle's identity **VID** correctly with non-negligible probability.

Definition 7 (Traceability:). We say that an anonymous communication scheme for cloud-based VANETs achieves traceability if STC can extract sender's identity form any valid ciphertext with non-negligible probability.

5 The Proposed Scheme

In this paper, we design a traceable and revocable attribute-based signcryption scheme. In order to realize efficient revocation, we employ an efficient binary tree algorithm to reduce the size of key updates from the traditional linear to the logarithmic level without expanding the key size. In order to support traceability, we insert the hidden identity into our ABSC scheme such that STC can directly capture traitor from the hidden identity information instead of querying each record in the unrevoked vehicle list as the conventional solutions do. In addition, we solve the key escrow problem that perplexes all existing attribute-based signcryption schemes.

Further, we propose the traitor-resistant and dynamic anonymous communication scheme based on the designed

ABSC. Specifically, the proposed scheme consists of the following eight algorithms.

- ***Setup*** (1^λ): The goal of this algorithm is to generate the system public parameter and the master private key which are necessary for the subsequent algorithm.
  - Define a Lagrange coefficient $\Delta_i^\gamma(x) = \prod_{j\in\gamma}^{j\neq i} \frac{x-j}{i-j}$ for $i \in Z_p$, where γ denotes a set of elements in Z_p. We can use the Lagrange coefficient to back out a polynomial over with an order $d-1$ by computing $q(x) = \sum_{i\in\gamma} q(i)\Delta_{i,\gamma}(x)$, where d denotes the threshold value. Define the attribute space U with $|\mathrm{U}| = n$ and the dummy attributes set Ω with $|\Omega| = \mathrm{d}$. Each attribute of $\mathrm{U}\cup\Omega$ corresponds to one element of Z_p. Let l_m denote the length of message m.
  - On input the security parameter λ, STC does as follows. Let $\hat{e} : G \times G \to G_1$ be a bilinear pairing, where G and G_1 are groups of a prime order p, and g denotes a generator of group G. Then, STC builds a binary tree. All vehicles in the system correspond to a leaf node respectively, and each node θ stores a random number KEK_θ. And each vehicle maintains the random number on the path nodes from the leaf node to the root. For instance, as shown in Fig.2, the path key of the vehicle u_2 is $PathKey = \{KEK_9, KEK_4, KEK_2, KEK_1\}$.

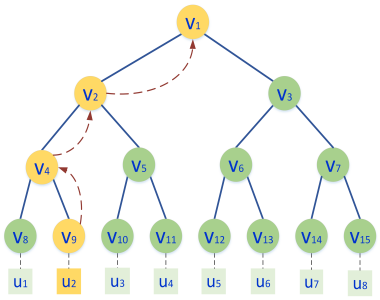

Fig. 2: The binary tree of the proposed scheme

  - STC selects $\alpha \leftarrow Z_p^*$ randomly. Then, STC sets $g_1 = g^\alpha$ and computes $Y = \hat{e}(g_1, g) = \hat{e}(g, g)^\alpha$. STC randomly chooses $t_1, t_2, ..., t_{n+1} \leftarrow G$ and sets $N = \{1, 2, ..., n+1\}$. $H_1 : \{0,1\}^* \to \{0,1\}^{l_m}$ and $H_2 : \{0,1\}^* \to G$ are two cryptography hash functions. STC, then, defines a function T: $T(x) = g^{x^n} \prod_{i=1}^{n+1} t_i^{\Delta_{i,N(x)}}$.
  - Finally, STC publishes the public parameter $mpk = (g, g_1, t_1, t_2, ..., t_{n+1}, H_1, H_2, Y)$, and holds the master private key $msk = \alpha$ on its own.
- ***KeyGen*** (mpk, msk, **A**): The goal of this algorithm is to generate the attribute-key for the vehicle. Specifically, this algorithm does as follows.
  - The vehicle **VID** randomly selects $\beta \leftarrow Z_p^*$ as its private user-key sk_{vid}, where **VID** denotes the real identity of the vehicle and **VID** $\in G$. Then, this vehicle computes the corresponding public user-key $pk_{vid} = g^\beta$. Finally, the vehicle sends the public user-key to STC and prove to STC that it owns the private user-key β by using zero knowledge technology.
  - After receiving the message from the vehicle, STC adds (**VID**, $pk_{vid,\mathbf{A}}$) to the vehicle list L_v, and randomly selects a polynomial $q(x)$ of order $d-1$ such that $q(o) = \alpha$. Then STC randomly chooses $r_1, r_2, ..., r_n \leftarrow Z_p$. Let **A** denote the attributes set owned by the vehicle **VID**.
  - Finally, STC returns the attribute-key $sk_{vid,\mathrm{A}} = (D_i = g^{\beta q(i)} T(i)^{r_i}, d_i = g^{r_i})_{i\in\mathrm{A}}$ together with the corresponding path key *PathKey* to the vehicle **VID**.
- ***KeyUpdate***(msk, G_i): The goal of this algorithm is to generate the key update information for unrevoked vehicles. STC periodically distributes key update messages to unrevoked vehicles and enforces this action when a vehicle exits the system. Before distributing the key update message, in order to prevent the revoked vehicles or other illegal users from getting this update message, STC needs to encrypt it, and only the unrevoked vehicles can decrypt it. Next, STC does as follows.
  - Let G_i denote the set of the unrevoked vehicles. For instance, the system has eight vehicles $\{u_1, u_2, u_3, u_4, u_5, u_6, u_7, u_8\}$, vehicles u_5 and u_6 exit the system, so that $G_i = \{u_1, u_2, u_3, u_4, u_7, u_8\}$.
  - In order to prevent the vehicles not belonging to G_i, that is to say, the revoked vehicles from being able to decrypt the key update message, STC selects the set of nodes in binary tree that can be minimally overwritten all nodes in G_i as $KEK(G_i)$. For instance, when $G_i = \{u_1, u_2, u_3, u_4, u_7, u_8\}$, as shown in Fig.3, the $KEK(G_i) = \{KEK_2, KEK_7\}$.

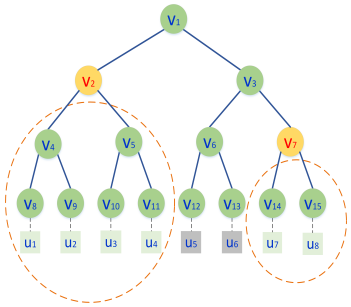

Fig. 3: The way of generating $KEK(G_i)$.

  - Finally, STC encrypts the key update information by using the random numbers correspond to the nodes in $KEK(G_i)$ respectively. In order to formalize the expres-

sion, we use *keyup* to represent the encrypted key update message, and $keyup = \{Enc_K(g^{\alpha'-\alpha})\}_{k \in KEK(G_i)}$, where α' is a random number selected in Z_p^*, Enc_K denotes a symmetric encryption algorithm, and K denotes the encryption key. As in the example above, $keyup = \{Enc_{KEK_2}(g^{\alpha'-\alpha}), Enc_{KEK_7}(g^{\alpha'-\alpha})\}$. STC publishes the new $Y = \hat{e}(g, g^{\alpha'})$.

- ***SigncryKG*** ($sk_{vid,\mathrm{A}}$, *keyup*, *PathKey*): The goal of this algorithm is to generate the updated attribute-key for the unrevoked vehicles. Unrevoked vehicles can decrypt the key update message by using the stored path key *PathKey*, and then can gain the new attribute key *newkey* by computing

$$\begin{aligned} D'_i &= D_i \cdot g^{\alpha'-\alpha} \\ &= g^{q(i)} \cdot (T(i))^{r_i} \cdot g^{\alpha'-\alpha} \\ &= g^{q(i)+\alpha'-\alpha} \cdot (T(i))^{r_i} \\ &= g^{q'(i)} \cdot (T(i))^{r_i} \end{aligned}$$

  where $q'(0) = \alpha'$.
- ***Signcrypt*** (w_e,$\Gamma_{k,S}$, $sk_{vid,A}$, $pk_{vid,A}$, *newkey*, β, m, t, **VID**): The goal of this algorithm is to generate ciphertext for the unrevoked vehicles. For the claim predicate $\Gamma_{k,S}$, the vehicle **VID** randomly selects a subset $S' \subset \{\mathrm{A} \cap S\}$ with $|S'| = k$, and a dummy attributes set $\Omega' \in \Omega$ with $|\Omega'| = d - k$. Then the vehicle chooses $r \leftarrow Z_p^*$ randomly, and computes $\sigma_1 = g^r$, $AID_1 = \mathbf{VID} \oplus H_2(g_1{}^r \cdot H_2(t))$, $AID_2 = \mathbf{VID}^{\beta}$, $AID_3 = \mathbf{VID} \oplus pk_{vid,\mathbf{A}}{}^r$, $\sigma_2 = \prod_{i \in \{S' \cup \Omega'\}} D_i{}^{-\beta} \cdot H_2(AID||T||m)^r$, $\sigma_3 = \prod_{i \in \{S' \cup \Omega'\}} (g^{r_i})^{-\beta}$ and $\sigma_4 = \prod_{i \in w_e} (T(i))^r$, where $AID = \{AID_1, AID_2, AID_3\}$. Then, the vehicle computes $K_e = Y^r$ and $c = H_1(K_e) \oplus m$. Finally, the vehicle broadcasts the ciphertext $C = \{w_e, \Gamma_{k,S}, \sigma_1, \sigma_2, \sigma_3, \sigma_4, AID, t\}$ where t is the timestamp.
- ***Unsigncrypt*** (w_e,$\Gamma_{\mathrm{k,S}}$, $\mathrm{sk}_{\mathrm{vid}',\mathbf{A}_\mathrm{r}}$, *mpk*, β', t): The goal of this algorithm is to decrypt ciphertext for the unrevoked vehicles. Assume the attributes set of the receiver is $\mathbf{A}_r$. When the intersection of set $\mathbf{A}_r$ and set w_e contains more elements than d, the receiver can decrypt the ciphertext. The receiver randomly selects a subset $D' \subset (\mathbf{A}_r \cap w_e)$ with $|D'| = d$. If there is no such set D', it returns $\perp$. Next, the receiver computes $K_e = \left(\frac{\hat{e}(\prod_{i \in D'} D_i{}^{-\beta'}, \sigma_1)}{\hat{e}(\prod_{i \in D'} d_i{}^{-\beta'}, \sigma_4)} \right)^{\Delta_{i,D'(0)}}$, where β' is the private user-key of the receiver. Then it can get the plaintext $m = H_1(K_e) \oplus c$. Finally, the receiver checks the following equation holds or not.

$$\left(\frac{\hat{e}(\sigma_2, g)}{\hat{e}(\prod_{i \in \{S' \cup \Omega'\}} T(i), \sigma_3) \cdot \hat{e}(H_2(AID||t||m), \sigma_1)}\right)^{\Delta_{i,S' \cup \Omega}(x)} = Y.$$

  If the above equation holds, the receiver accepts the message m. Otherwise, it rejects.
- ***Revoke*** (t, ***VID***)***:*** The goal of this algorithm is to achieve the vehicle revocation. If the vehicle **VID** wants to exist the system, it sends the identity **VID** to STC. Then STC deletes the corresponding leaf node in the set G_i, and updates G_i. Finally, the algorithm outputs the new set G_i.
- ***Trace*** (t, AID)***:*** The goal of this algorithm is to achieve traceability of traitor. When a vehicle transmits incorrect information or has some illegal operation, under the condition of protecting the privacy of the vehicle's identity, STC can capture traitor. Concretely, STC computes $\mathbf{VID} = AID_1 \oplus H_2({\sigma_1}^{\alpha}, H_2(t))$, $pk_{vid,\mathbf{A}} = \mathbf{VID} \oplus AID_3$. Then, STC checks whether the equation $\hat{e}(\mathbf{VID}, pk_{vid,\mathbf{A}}) = \hat{e}(AID_2, g)$ holds or not. If the above equation holds, STC checks whether there exists a $pk_{vid,\mathbf{A}}$ in the vehicle list L_v. If so, STC returns traitor's identity **VID**. As we can see, identity **VID** can only be obtained with a master key α, so only STC can trace back to the vehicle.

6 Security Analysis

In this section, we analyze the security of our proposed scheme in term of confidentiality of driving data, unforgeability, anonymity and tracebility.

Theorem 1. (***Confidentiality of Driving Data***) Suppose there exists no polynomial algorithm that can solve the DBDH problem in bilinear groups. An adversary, who does not own a set of attributes satisfying the access control structure of the ciphertext, can decrypt the ciphertext correctly with negligible probability.

Proof. To prove this theorem, we define a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Game 1: In the Game 1, both the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ perform as defined in the security definition. That is, the Challenger $\mathcal{C}$ runs the ***Setup*** algorithm to generate the master key *msk*, the system public parameter *mpk*. Then the challenger $\mathcal{C}$ sends the public parameter *mpk* to the adversary $\mathcal{A}$. Next, the adversary $\mathcal{A}$ does as **Query Phase**. After that, the adversary $\mathcal{A}$ randomly selects two equal length messages m_0 and m_1, and sends these messages to the challenger $\mathcal{C}$. Then, the challenger generates the ciphertext m_b by running ***Signcrypt***, where $b \leftarrow \{0, 1\}$. Finally, after receiving the challenged ciphertext C from $\mathcal{C}$, the adversary $\mathcal{A}$ guesses a bit b' for b.

Analysis: Assume that the adversary $\mathcal{A}$ wins the Game 1 with non-negligible probability. Then, we can construct a simulator $\mathcal{T}$ to solve the DBDH problem. The simulator is given $g_1 = g^x$ and $g_2 = g^y$, where g^y is the public user-key. The simulator $\mathcal{T}$ acts like the challenger $\mathcal{C}$ in Game 1.

- **Setup Phase**: Firstly, the adversary $\mathcal{A}$ selects the specified attributes set w_e^* and sends it to $\mathcal{T}$. Then, the simulator $\mathcal{T}$ performs the ***Setup*** algorithm and returns the common parameters $mpk = (g, g_1, t_1, t_2, ..., t_{n+1}, H_1, H_2, Y)$ to $\mathcal{A}$.
- **Query Phase**: The adversary $\mathcal{A}$ makes inquiries at most polynomial times, and does as the security definition.

- **Challenge Phase**: The adversary $\mathcal{A}$ outputs two challenge messages m_0^*, m_1^* and the claim predicate $\Gamma_{k,S}^*$. The simulator $\mathcal{T}$ randomly selects $b \leftarrow \{0,1\}$. Then it runs the ***KeyGen*** algorithm to get the signcryption key. The simulator $\mathcal{T}$ randomly selects $r \leftarrow Z_p^*$, and then calculates k_e^*=h^r, c_b^*=$\mathrm{H}_1(k_e^*) \oplus m_b^*$, σ_1^*=g^{zr}, σ_2^*=$\prod_{\mathrm{i}\in\{\mathrm{S'UU'}\}} \mathrm{D}_i{}^{-y} \cdot \mathrm{H}_2(\mathrm{AID}||\mathrm{t}||m)^r$, $\sigma_3^* = \prod_{i\in\{S'\cup\Omega'\}} (g^{r_i})^{-y}$, $AID_1^* = VID^* \oplus H_2({g_1}^r \cdot H_2(t^*))$, $AID_2^* = \mathbf{VID}^y$, $AID_3^* = \mathbf{VID}^* \oplus pk_{vid,\mathcal{A}}$ and $\sigma_4^* = \prod_{i\in w_e} (T(i))^{zr}$, where $AID^* = \{AID_1^*, AID_2^*, AID_3^*\}$. Next, the simulator $\mathcal{T}$ sends the ciphertext $C^* = \{w_e^*, \Gamma_{k,S}^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, AID^*, t^*\}$ to the adversary $\mathcal{A}$.
- **Guess Phase**: Firstly, the adversary $\mathcal{A}$ makes inquiries as in **Query Phase**. But when $|w_e^* \cap w_r| \geq d$, the adversary $\mathcal{A}$ is not allowed to make *keyGen* queries and *Unsigncryption* queries on the attributes set w_r. Finally, the adversary $\mathcal{A}$ returns a bit b'.

When $h = \hat{\mathrm{e}}(g,g)^{xyz}$, C^* is a valid ciphertext. The reason is that if the receiver's attributes set is w_r, we randomly selects a subset $\mathrm{D}' \subset w_e^* \cap w_r$ with $|\mathrm{D}'| = d$, and then we can get

$$\begin{aligned} k_e^* &= \left(\frac{\hat{e}(\prod_{i\in \mathrm{D}'} \mathrm{D}_i{}^{-y}, \sigma_1^*)}{\hat{e}(\prod_{i\in \mathrm{D}'} d_i{}^{-y}, \sigma_4^*)} \right)^{\Delta_{i,\mathrm{D}'(0)}} \\ &= \left(\frac{\hat{e}(\prod_{i\in \mathrm{D}'} g^{-q(\mathrm{i})\cdot y} \mathrm{T}(i)^{-r_i y}, g^{zr})}{\hat{e}(\prod_{i\in \mathrm{D}'} g^{-r_i y}, \mathrm{T}(i)^{zr})} \right)^{\Delta_{i,\mathrm{D}'(0)}} \\ &= \hat{\mathrm{e}}(g_1, g_2)^{zr} \\ &= \hat{\mathrm{e}}(g,g)^{xyzr} \end{aligned}$$

So, the algorithm $\mathcal{T}$ determines whether h is equal to $\hat{\mathrm{e}}(g, g^{xyz})$ based on whether the adversary $\mathcal{A}$ wins the game. Furthermore, the DBDH problem can be solved.

In order to ensure the victory of the adversary $\mathcal{A}$, according to [8], all the selected w_r sets need to satisfy $|w_r \cap w_e^*| \geq d$. The probability is

$$p_1 = \frac{\mathrm{C}_n^d + \mathrm{C}_n^{d+1} + ... + \mathrm{C}_n^m}{\mathrm{C}_n^0 + \mathrm{C}_n^1 + ... + \mathrm{C}_n^n} = \frac{\mathrm{C}_n^d + \mathrm{C}_n^{d+1} + ... + \mathrm{C}_n^m}{2^n} < 1.$$

And the probability is

$$p_2 = 1 - \frac{q_k}{\mathrm{C}_n^0 + \mathrm{C}_n^1 + ... + \mathrm{C}_n^n} = 1 - \frac{q_k}{2^n}$$

for w_r not satisfies the claim predicate $|w_r \cap w_e^*| \geq d$. Moreover, the adversary $\mathcal{A}$ queries to *KeyGen*, *Signcrypt*, *Unsigncrypt* phases are q_k, q_u, q_{us} times respectively with the advantage ε. Thus, the probability of breaking the DBDH problem is $\varepsilon' = \frac{\varepsilon \cdot p_1 p_2}{q_k q_u q_{us}} < \frac{\varepsilon \cdot (1 - \frac{q_k}{2^n})}{q_k q_u q_{us}}$.

In conclusion, due to the difficulty of DBDH, we can obtain that our proposed scheme satisfies the confidentiality of driving data against selective ciphertext attacks.

Theorem 2. (***Unforgeability***) Suppose there exists no polynomial algorithm that can solve CDH problem and DL problem in bilinear groups. An adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$, in the proposed scheme, cannot forge a valid ciphertext for other vehicles with non-negligible probability.

Proof. To prove this theorem, we define two games between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}_1$, and between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}_2$ respectively.

Game 2: In the Game 2, both the challenger $\mathcal{C}$ and the adversary $\mathcal{A}_1$ perform as defined in the security definition. That is, the Challenger $\mathcal{C}$ runs the ***Setup*** algorithm to generate the master key *msk*, the system public parameter *mpk*. Then the challenger $\mathcal{C}$ sends the public parameter *mpk* to the adversary $\mathcal{A}_1$. Next, the adversary $\mathcal{A}_1$ does as **Query Phase**. After that, the adversary $\mathcal{A}_1$ generates a forged ciphertext C^*.

Analysis: Assume that the adversary $\mathcal{A}_1$ wins the Game 2 with non-negligible probability. Then, we can construct a simulator $\mathcal{F}$ to solve the CDH problem. The simulator $\mathcal{F}$ acts like the challenger $\mathcal{C}$ in Game 2.

- **Setup Phase:** The adversary $\mathcal{A}_1$ selects a signature claim predicate $\Gamma_{k,S}^*$, and sends it to $\mathcal{F}$. Then, $\mathcal{F}$ sets $g_1 = g^x$ and $g_2 = g^y$, where g^y is the public user-key. Next, $\mathcal{F}$ randomly selects polynomial functions $f(x)$ and $v(x)$ with the degree n. And if x satisfies $\Gamma_{k,S}^*$, $v(x) = -x^n$. Otherwise, $v(x) \neq -x^n$. And for $\forall x \in [1, n+1]$, $t_i = g^{v(i)} g^{f(x)}$. Then, we can get $T(x) = g^{x^n + v(x)} g^{f(x)}$. The reason is that

$$\begin{aligned} T(x) &= g^{x^n} \prod_{i=1}^{n+1} {t_i}^{\Delta_{i,\mathrm{N}(x)}} \\ &= g^{x^n} \prod_{i=1}^{n+1} (g^{v(i)} \cdot g^{f(i)})^{\Delta_{i,\mathrm{N}(x)}} \\ &= g^{x^n} g^{\sum_{i=1}^{n+1} v(i)\Delta_{i,\mathrm{N}(x)}} g^{\sum_{i=1}^{n+1} f(i)\Delta_{i,\mathrm{N}(x)}} \\ &= g^{x^n + v(x)} g^{f(x)} \end{aligned}$$

  Next, the simulator $\mathcal{F}$ randomly selects $\gamma \leftarrow Z_p$. Let $\mathrm{H}_2(\mathrm{AID}||\mathrm{t}||m) = \mathrm{g}^\gamma$. The algorithm $\mathcal{F}$ sends the public parameter $mpk = (g, g_1, t_1, t_2, ..., t_{n+1}, H_1, H_2, Y)$ to $\mathcal{A}_1$.
- **Query Phase:** The adversary $\mathcal{A}_1$ makes queries to the challenger $\mathcal{F}$ as in security definition.
- **Forgery Phase:** Finally, the adversary $\mathcal{A}_1$ outputs a forged ciphertext $C^* = \{w_e, \Gamma_{k,S}^*, \sigma_1, \sigma_2, \sigma_3, \sigma_4, AID, t\}$.

The ***Unsigncrypt*** algorithm takes the forged ciphertext C^* as the input. Then, we have $(\frac{\hat{e}(\sigma_2, g)}{\hat{e}(\prod_{i\in\{S'\cup\Omega'\}} T(i), \sigma_3) \cdot \hat{e}(H_2(AID||t||m), \sigma_1)})^{\Delta_{i,S'\cup\Omega}(x)} = \hat{e}(g_1, g_2)$. Thus, C^* is a valid ciphertext. Due to $\mathrm{H}_2(\mathrm{AID}||\mathrm{t}||m) = \mathrm{g}^\gamma$ and $T(x) = g^{x^n + v(x)} g^{f(x)}$ for all x satisfies $\Gamma_{k,S}^*$, the above equation is equivalent to the following expression.

$$\begin{aligned} &(\frac{\hat{e}(\sigma_2, g)}{\hat{e}(\prod_{i\in\{S'\cup\Omega'\}} T(i), \sigma_3) \cdot \hat{e}(H_2(AID||t||m), \sigma_1)})^{\Delta_{i,S'\cup\Omega}(x)} \\ &= (\frac{\hat{e}(\sigma_2, g)}{\hat{e}(\prod_{i\in\{S'\cup\Omega'\}} {\sigma_3}^{f(i)}, T(i)) \cdot \hat{e}(g, {\sigma_1}^\gamma)})^{\Delta_{i,S'\cup\Omega}(x)} \\ &= \hat{e}(\frac{\sigma_2}{{\sigma_3}^{f(i)} {\sigma_1}^\gamma}, g)^{\Delta_{i,S'\cup\Omega}(x)} \end{aligned}$$

Since,

$$(\frac{\hat{e}(\sigma_2,g)}{\hat{e}(\prod\limits_{i\in\{S'\cup\Omega'\}} T(i),\sigma_3)\cdot\hat{e}(H_2(AID||t||m),\sigma_1)})^{\Delta_{i,S'\cup\Omega}(x)}$$
$$=\hat{e}(g_1,g_2)=\hat{e}(g,g^{xy})$$

We get

$$\hat{e}(\frac{\sigma_2}{{\sigma_3}^{f(i)}{\sigma_1}^{\gamma}},g)^{\Delta_{i,S'\cup\Omega}(x)}=g^{xy}.$$

Thus, the algorithm $\mathcal{F}$ can calculate g^{xy}. Furthermore, the CDH problem can be solved.

The advantage of the adversary $\mathcal{A}_1$ is $p_1 = \frac{C_n^d}{C_n^0+C_n^1+...+C_n^n} = \frac{n!}{(n-d)!d!2^n}$. Since $\mathcal{A}_1$ is not allowed to make *Signcrypt* query on $(m, w_e^*, \Gamma_{k,S}^*)$. And the probability is $p_2 = 1 - \frac{q_S}{C_n^0+C_n^1+...+C_n^n} = 1 - \frac{q_S}{2^n}$.

Moreover, the times of adversary $\mathcal{A}_1$ queries to *KeyGen*, *Signcrypt*, *Unsigncrypt* algorithms are q_k, q_u, q_{us} respectively with the advantage ε. Thus, the probability of breaking the CDH problem is

$$\varepsilon' = \frac{\varepsilon\cdot p_1p_2}{q_kq_uq_{us}} < \frac{\varepsilon\cdot n!\,(1-\frac{q_S}{2^n})}{(n-d)!\,d!\,2^nq_kq_uq_{us}}.$$

In conclusion, due to the difficulty of CDH, we can obtain that our proposed scheme satisfies unforgeability against the chosen message attack.

Similarly, we define a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}_2$, who owns the master private key.

Game 3: In the Game 3, both the challenger $\mathcal{C}$ and the adversary $\mathcal{A}_2$ perform as defined in the security definition. That is, the Challenger $\mathcal{C}$ runs the ***Setup*** algorithm to generate the master key *msk*, the system public parameter *mpk*. Then the challenger $\mathcal{C}$ sends the public parameter *mpk* and the the master key *msk* to the adversary $\mathcal{A}_2$. Next, the adversary $\mathcal{A}_2$ does as **Private User-Key Phase**. After that, the adversary $\mathcal{A}_2$ generates a forged ciphertext C^*.

Analysis: Assume that the adversary $\mathcal{A}_2$ wins the Game 3 with non-negligible probability. Then, we can construct a simulator $\mathcal{F}^*$ to solve the Discrete Logarithm (DL) problem. The simulator $\mathcal{F}^*$ acts like the challenger $\mathcal{C}$ in Game 3.

- **Setup Phase:** The simulator $\mathcal{F}^*$ performs the ***Setup*** algorithm, and sends the common parameters $mpk = (g, g_1, t_1, t_2, ..., t_{n+1}, H_1, H_2, Y)$ and the master private key *msk* to $\mathcal{A}_2$.
- **Private User-Key Queries:** The the adversary $\mathcal{A}_2$ queries the private user-key except for the vehicle VID^*. Then the simulator $\mathcal{F}^*$ randomly selects $\beta \leftarrow Z_p^*$, and returns it to $\mathcal{A}_2$.
- **Forgery Phase:** The adversary $\mathcal{A}_2$ generates a forged ciphertext C^* on the time period tp^*, the attributes set w_e^*, the claim predicate $\Gamma_{k,S}^*$ and the message m^*.

If the forged ciphertext C^* is valid, β^* is the private user-key of vehicle VID^*, where $\sigma_3^* = \prod\limits_{i\in\{S'\cup\Omega'\}} (g^{r_i})^{-\beta^*}$. We can know that given the public user-key g^{β^*} of vehicle VID^*, the simulator $\mathcal{F}^*$ can extract β^*, i.e., the simulator $\mathcal{F}^*$ breaks Discrete Logarithm problem. In conclusion, due to the difficulty of DL, we can obtain that our proposed scheme satisfies unforgeability for adversary $\mathcal{A}_2$.

In summary, our proposed scheme satisfies unforgeability.

Theorem 3. (***Anonymity***) The valid receivers cannot derive the identity information of sender, besides the malicious adversaries cannot derive any identity information of sender in our scheme.

Proof. To prove this theorem, we define a game between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

Game 1: In the Game 1, both the challenger $\mathcal{C}$ and the adversary $\mathcal{A}$ perform as defined in the security definition. That is, the Challenger $\mathcal{C}$ runs the ***Setup*** algorithm to generate the master key *msk*, the system public parameter *mpk*. Then the challenger $\mathcal{C}$ sends the public parameter *mpk* to the adversary $\mathcal{A}$. Next, the adversary $\mathcal{A}$ does as **Query Phase**. Finally, the adversary $\mathcal{A}$ outputs the vehicle's identity VID*.

Analysis: Assume the vehicle **VID** possessed the attributes set **A** generates a ciphertext $C^* = \{w_e^*, \Gamma_{k,S}^*, \sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, AID^*, t^*\}$, where $\mathbf{A}\cap S > k$. We can see that σ_1^* and t^* are independent of the set **A** and the vehicle identity **VID**. Also, because $AID_1^* = \mathbf{VID}^* \oplus H_2({g_1}^r \cdot H_2(t^*))$, $AID_2^* = \mathbf{VID}^\beta$, $AID_3^* = \mathbf{VID}^* \oplus pk_{vid,\mathbf{A}}$, $\sigma_2^* = \prod\limits_{i\in\{S'\cup\Omega'\}} {D_i}^{-\beta}\cdot H_2(AID^*||t^*||m)^r$, $\sigma_3^* = \prod\limits_{i\in\{S'\cup\Omega'\}} (g^{r_i})^{-\beta}$ and $\sigma_4^* = \prod\limits_{i\in w_e} (T(i))^r$, where β and r are random numbers in Z_p^*, $(\sigma_1^*, \sigma_2^*, \sigma_3^*, \sigma_4^*, AID^* = \{AID_1^*, AID_2^*, AID_3^*\})$ has a uniform distribution over the group G. In conclusion, the ciphertext will not reveal any information about the sender's attributes set **A** and the identity of the vehicle. Therefore, our proposed scheme achieves anonymity.

Theorem 4. (***Tracebility***) Only STC can trace back to the corresponding sender by the hidden identity included in the ciphertext.

Proof. This can be done directly by ***Trace*** algorithm. Concretely, STC computes $\mathbf{VID} = AID_1 \oplus H_2({\sigma_1}^\alpha, H_2(t))$, $pk_{vid,\mathbf{A}} = \mathbf{VID} \oplus AID_3$. Then, STC checks whether the equation $\hat{e}(\mathbf{VID}, pk_{vid,\mathbf{A}}) = \hat{e}(AID_2, g)$ holds or not. If the above equation holds, STC checks whether there exists a $pk_{vid,\mathbf{A}}$ in the vehicle list L_v. If so, STC returns traitor's identity **VID**. As we can see, identity **VID** can only be obtained with a master key α, so only STC can trace back to the vehicle. In short, our scheme can complete the traceability of the vehicle under the condition of protecting the privacy of vehicle's identity.

7 Performance and Experiments

In this section, we first give functionality and efficiency comparison among our scheme and several related schemes, and the comparison between our scheme and scheme [8]. Then, we evaluate the performance of our scheme in experiments.

7.1 Performance Evaluation

In this section, we first compare TD-ACF with several related schemes [8–10, 28, 32] in terms of functionality, the efficiency of traceability, and the size of key updates. Secondly, we compare TD-ACF with scheme [8] in terms of the size of attribute-key, the size of key updates and the

TABLE 2: **Comparison of functionality and efficiency among our scheme and schemes [8–12, 28, 32], where $\bar{\surd}$ denotes that it is not fully achieved, and R denotes the number of revoked vehicles**

Schemes	Confidentiality	Authentication	Anonymity	Traceability	Efficiency of Traceability	Size of Key Update
[28]	×	√	√	×	-	*linear*
[32]	×	√	√	$\bar{\surd}$	-	*linear*
[9]	√	×	√	×	-	*linear*
[8]	×	√	√	√	*O*(R)	*logarithmic*
[10]	√	√	√	×	-	*linear*
[11]	√	√	√	×	-	*linear*
[12]	√	√	√	×	-	*linear*
Ours	√	√	√	√	*O*(1)	*logarithmic*

TABLE 3: **The comparison among our scheme and scheme [8]**

	Size of Attribute-Key	Size of Key Updates	Efficiency of Traceability
The scheme [8]	$(\|\mathbf{A}\|+d)\cdot\log(N+1)$	$R\log(\frac{N}{R})$ or (N-R)	*O*(R)
Our scheme	$(\|\mathbf{A}\|+d)+\log(N+1)$	$R\log(\frac{N}{R})$ or (N-R)	*O*(1)

TABLE 4: **The computation overhead of the proposed revocable and traceable attribute-based signcryption scheme**

	Our scheme
Signcryption	$(3(k+d)+2)Exp_G+3Hash+3(k+d)Mul_G+xor$
Unsigncryption	$3d\cdot Exp_{G_1}+5Pair+Hash+xor+3Mul_{G_1}$

efficiency of traceability. Finally, we analyze the proposed scheme qualitatively.

Before the comparison, we set a bilinear pairing $\hat{e}: G\times G\rightarrow G_1$, and define the following notations to denote the operations in our scheme. Let $Hash$, $Exp(G)$ and $Mul(G)$ respectively denote one hashing operation, one exponentiation operation in G and one multiplication operation in G. $Exp(G_1)$ and $Mul(G_1)$ denote one exponentiation operation and one multiplication operation in G_1 respectively. Similarly, $Mul(Z_p^*)$ denotes one addition operation in Z_p^*. *Pair* and *xor* denote one pairing operation and one XOR operation respectively. Suppose the set of attributes is **A**, and $|\mathbf{A}|$ denotes the number of the elements included in this set. Let *d* and *k* denote predefined size of the default attributes set and the number of attributes used for signcrypt respectively. And N and R respectively denote the number of all the vehicles and the revoked vehicles in the system.

Comparison of Functionality and Efficiency. As shown in Table 2, in terms of functionality, schemes [8], [28] and [32] cannot satisfy the confidentiality of driving data. Schemes [9], [10], [11], [12] and [28] cannot support traceability. In addition, schemes [9] and [10] cannot support authentication. Our scheme is the only scheme that can satisfy all of the following properties: confidentiality of driving data, authentication, anonymity and traceability. In aspect of efficiency, STC needs to issue a new key for each unrevoked vehicle for key update, so the size of key update is linear to the number of unrevoked vehicles in schemes [9], [10], [11], [12], [28] and [32]. The size of key update, in our scheme and scheme [8], is logarithmic to the number of unrevoked vehicles. The computation complexity of traceability is *O*(R) in scheme [8]. However, it is *O*(1) in our scheme. Therefore, the above comparison shows that our scheme meets more properties while is more efficient.

Comparison between Our Scheme and Scheme [8].

In a nutshell, our scheme has the following advantages over the scheme in [8]: 1) In our scheme, the Semi-Trusted Cloud (STC) can precisely capture and punish traitor directly, instead of storing a list of unrevoked vehicles and querying each record in the list does as the scheme in [8]. 2) The scheme in [8] does not take into account the confidentiality of driving data. Our scheme supports anonymous authentication and confidentiality of driving data simultaneously. 3) In addition, the scheme in [8] has a security weakness. The Trusted Authority (TA) can impersonate any vehicle to generate a valid signature under certain conditions. 4) The most important one is that, in [8], a binary tree structure is incorporated into the key generation and key update processes to support the revocation of vehicle. The resulting long-term attribute-key size is large, which incurs a heavy storage burden to the resource-constrained vehicle. In our scheme, the size of private attribute-key is small by using the new binary tree. Specifically, the comparison of attribute-key size is described as follows:

As shown in Table 3, in scheme [8], the size of private attribute-key is $(|\mathbf{A}|+d)\cdot\log(N+1)$, which is linear with the number of the (default) attributes possessed by a vehicle and the height of binary tree. In our scheme, the size of private attribute-key is $(|\mathbf{A}|+d)+\log(N+1)$. In the general case, the inequality $N\geq|\mathrm{A}|+\mathrm{d}$ is true. Therefore, the size of private attribute-key in scheme [8] is a multiple of that in our scheme. The size of key updates in scheme [8] and our scheme have been reduced to the logarithmic level, with the specific value of $R\log(\frac{N}{R})$. STC needs to query each record in the list of unrevoked vehicles one by one to capture traitor in scheme [8]. Therefore, in scheme [8], the efficiency of traceability is linear with the number of unrevoked vehicles in the system. However, in our scheme, STC can directly capture traitor without the need for an unrevoked vehicles list. From the above analysis, our scheme is superior to the scheme [8] in terms of attribute-key size and the efficiency of traceability.

Qualitative Analysis of Our Scheme. In our scheme, each sender performs $3(k+d)+2$ exponentiation operations, $3(k+d)$ multiplication operations in group G, three hash operations and one XOR operation to generate a ciphertext. And, each receiver performs $3d$ exponentiation operations, 3 multiplication operations in group G_1, five pair operations, one hash operation and one XOR operation to decrypt the ciphertext.

7.2 Experimental Results

In this section, we evaluate the performance of our proposed scheme in several experiments. We run these experiments on a Linux server with Intel processor running at 2.30 GHz and 8 GB memory. With the help of Pairing-Based Cryptography (PBC) library [37] and the GNU Multiple Precision Arithmetic (GMP) [38], we utilize C++ language to implement the simulations. In the experiments, we choose a bilinear map that uses a supersingular curve to achieve the fast pairing operations. We set the base field size to be 512bits, the size of an element in Z_q^* is 160bits.

Computation Overhead of Key Update. In the following experiments, we set the total number of vehicles to be $\mathrm{N} = 2^{10} = 1024$. And the size of the default attributes set is set to d=4 and the threshold is set to k=3. As shown in Fig.4, we first evaluate the computation time of ***KeyUpdate*** when the number of revoked vehicles ranges from 1 to 1024. Then we observe that when the number of revoked vehicles is less than $\frac{\mathrm{N}}{2} = 512$, its computation time is logarithmic to $\frac{\mathrm{N}}{\mathrm{R}}$. However, when the number of revoked vehicles is larger than $\frac{\mathrm{N}}{2} = 512$, the computation time is linear with (N-R). We also see that when the number of revoked vehicles varies from 1 to 1024, the computation time of ***KeyUpdate*** ranges from 0.043s to 4.4036s.

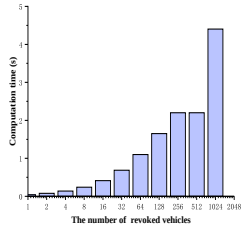

Fig. 4: The computation time of ***KeyUpdate***

Computation Overhead of Tracking Traitor. As shown in Fig.5, we evaluate the computation time of tracking traitor when the total number of vehicles ranges from 100 to 1024. In the scheme [8], the computation time of tracking traitor is linear with the total number of vehicles. Specifically, when the total number of vehicles ranges from 100 to 1024, the corresponding computation time varies from 0.523s to 5.33s. In the proposed scheme, the computation time of tracking traitor is a constant 0.0046s. Thus, our scheme is more efficient in tracking traitor than the scheme [8].

Computation Overhead of *Signcrypt* and *Unsigncrypt*. As shown in Fig.6 and Fig.7, we evaluate the computation time of algorithms ***Signcrypt*** and ***Unsigncrypt*** when the total number of ciphertexts ranges from 0 to 100. We can

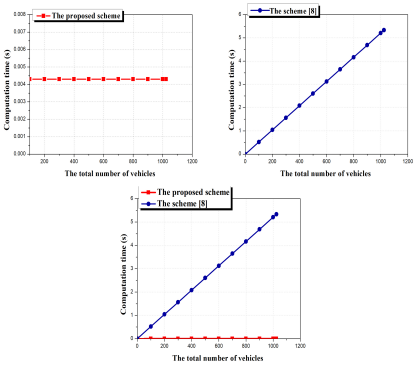

Fig. 5: The computation time of tracking traitor

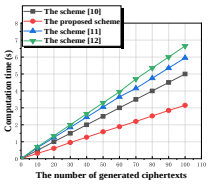

Fig. 6: The computation time of the algorithm ***Signcrypt***

see that, from above two pictures, the computation time of algorithms ***Signcrypt*** and ***Unsigncrypt*** is linear with the number ciphertexts. For ***Signcrypt*** process, we set the number of ciphertexts ranges from 0 to 100. Specifically, when the number of ciphertexts ranges from 0 to 100, in the scheme in [10], the corresponding computation time varies from 0s to 5.012s. In the scheme in [11], the corresponding computation time varies from 0s to 5.908s. When the number of ciphertexts ranges from 0 to 100, the corresponding computation time varies from 0s to 6.751s in the scheme in [12]. In our scheme, the corresponding computation time varies from 0s to 3.152s for the algorithm ***Signcrypt***. For ***UnSigncrypt*** process, we also set the number of ciphertexts ranges from 0 to 100. In the scheme [10], when the number of ciphertexts ranges from 0 to 100, the corresponding computation time varies from 0s to 6.05s. In the scheme in [11], the corresponding computation time varies from 0s to 8.03s. When the number of ciphertexts ranges from 0 to 100, the corresponding computation time varies from 0s to 9.13s in the scheme in [12]. In our scheme, the time varies from 0s to 4.883s for the algorithm ***Unsigncrypt***. Therefore, our scheme is more efficient in singncryption and unsigncryption than existing works [10, 11, 12].

8 Conclusion

In this paper, to provide a secure and practical anonymous communication service for cloud-based VANETs, we proposed a traitor-resistant and dynamic anonymous communication framework (TD-ACF) which supports confidentiality of driving data, anonymous authentication simultaneously. Moreover, TD-ACF supports two practical requirements in cloud-based VANETs: the revocation and the traceability of traitor. In our scheme, each vehicle is represented by a set of attributes instead of its real identity, so anyone cannot

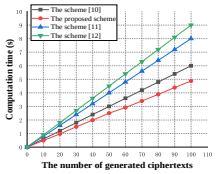

Fig. 7: The computation time of the algorithm ***Unsigncrypt***

know the real identity of the sender during communication. The security proof and the experimental analysis demonstrate that the proposed scheme achieves desirable security and efficiency. With the help of pre-computation, server-aided computation and cryptographic hardware, the performance of the proposed framework scheme can be greatly improved, and its practical applications are no longer a problem.

ACKNOWLEDGMENTS

We are grateful to the anonymous reviewers for their invaluable suggestions. This work was supported by the National Natural Science Foundation of China (Grant No. 61972094 and 62032005), and in part by the young talent promotion project of Fujian Science and Technology Association.

REFERENCES

[1] S. Horng, S. T, Y. P, P. F, et al., b-SPECS+: Batch verification for secure pseudonymous authentication in VANET, in *TIFS*, vol. 8, no. 11, pp. 1860-1875, 2013.

[2] K. A. Shim, CPAS: An efficient conditional privacy-preserving authentication scheme for vehicular sensor networks, in *TITS*, vol. 61, no. 4, pp. 1874-1883, 2012.

[3] J. K. Liu, T. H. Yuen, M. H. Au, and W. Susilo, Improvements on an authentication scheme for vehicular sensor networks, in *Expert Systems with Applications*, vol. 41, no. 5, pp. 2559-2564, 2014.

[4] M. Bayat, M. Barmshoory, M. Rahimi, and M. R. Aref, A secure authentication scheme for VANETs with batch verification, in *Wireless Network*, vol. 21, no. 5, pp. 1733-1743, 2015.

[5] J. Petit, F. Schaub, M. Feiri, and F. Kargl, Pseudonym schemes in vehicular networks: A survey, in *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1, pp. 228-255, 2015.

[6] C. Li, M. S. Hwang, and Y. Chu, A secure and efficient communication scheme with authenticated key establishment and privacy preserving for vehicular ad hoc networks, in *Computer Communications*, vol. 31, no. 12, pp. 2803-2814, 2008.

[7] A. Singh and H. C. S. Fhom, Restricted usage of anonymous credentials in vehicular ad hoc networks for misbehavior detection, in *International Journal of Information Security*, vol. 16, no. 2, pp. 195-211, 2017.

[8] H. Cui, R. Deng, and G. Wang, An Attribute-Based Framework for Secure Communications in Vehicular Ad Hoc Networks, in *IEEE/ACM Transactions on Networking*,, doi:10.1109/tnet.2019.2894625, 2019.

[9] D. Huang and M. Verma, ASPE: Attribute-based secure policy enforcement in vehicular ad hoc networks, in *Ad Hoc Network*, vol. 7, no. 8, pp. 1526-1535, 2009.

[10] Q. Kang, X. Liu, Y. Yao, Z. Wang, and Y. Li, Efficient authentication and access control of message dissemination over vehicular ad hoc network, in *Neurocomputing*, vol. 181, pp. 132-138, 2015.

[11] J. Kim, and J. Song, A Pre-Authentication Method for Secure Communications in Vehicular Ad Hoc Networks, presented at *8th International Conference on Wireless Communications, Networking and Mobile Computing*, 2012.

[12] P. Vijayakumar, M. Azees, A. Kannan, and L. Deborah, Dual Authentication and Key Management Techniques for Secure Data Transmission in Vehicular Ad Hoc Networks, in *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1015-1028, 2016.

[13] Y. S. Rao and R. Dutta, Efficient attribute based access control mechanism for vehicular ad hoc network, presented at *International Conference on Network and System Security*, pp. 26-39, 2013.

[14] M. Raya and J. Hubaux, Securing vehicular ad hoc networks, in *Journal of Computer Security*, vol. 15, no. 1, pp. 39-68, 2007.

[15] R. Lu, X. Lin, H. Zhu, P. Ho, and X. Shen, ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications, presented at *The 27th Conference on Computer Communications*, pp. 1903-1911, 2008.

[16] C. Zhang, X. Lin, R. Lu, and P. Ho, RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks, presented at *IEEE International Conference on Communications*, pp. 1451-1457, 2008.

[17] C. Zhang, R. Lu, X. Lin, P. Ho, and X. Shen, An efficient identity-based batch verification scheme for vehicular sensor networks, presented at *IEEE Infocom*, pp. 816-824, 2008.

[18] J. P. Hubaux, S. Capkun, and J. Luo, The security and privacy of smart vehicles, in *IEEE Security and Privacy*, vol. 2, no. 3, pp. 49-55, 2004.

[19] P. Kamat, A. Baliga, and W. Trappe, An identity-based security framework for VANETs, presented at *The 3rd International Workshop / Vehicular Ad Hoc Networks (VANET)*, pp. 94-95, 2007.

[20] G. Calandriello, P. Papadimitratos, J. Hubaux, and A. Lioy, Efficient and robust pseudonymous authentication in VANET, presented at *The 4th ACM international workshop on Vehicular ad hoc networks*, pp. 19-28, 2007.

[21] X. Lin, X. Sun, P. Ho, and X. Shen, GSIS: A secure and privacy-preserving protocol for vehicular communications, in *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442-3456, 2007.

[22] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J. P. Hubaux, Eviction of misbehaving and faulty nodes in vehicular networks, in *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1557-1568, 2007.

[23] A. Studer, E. Shi, F. Bai, and A. Perrig, TACKing together efficient authentication, revocation, and privacy in VANETs, presented at *The 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, pp. 1-9, 2009.

[24] A. Wasef and X. Shen, MAAC: Message authentication

acceleration protocol for vehicular ad hoc networks, in *Engine*, pp. 1-6, 2009.

[25] H. Wen, P. Ho, and G. Gong, A novel framework for mes-sage authentication in vehicular communication networks, presented at *IEEE Global Telecommunications Conference*, doi: 10.1109/GLOCOM.2009.5425519, 2009.

[26] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer, A scalable robust authentication protocol for secure vehicular communications, in *IEEE Transactions on Vehicular Technology*, vol. 59, no. 4, pp. 1606-1617, 2010.

[27] B. Qin, Q. Wu, J. Domingo-Ferrer, and L. Zhang, P-reserving security and privacy in large-scale VANETs, presented at *International Conference on Information and Communications Security*, pp. 121-135, 2011.

[28] D. Föster, F. Kargl, and H. Löhr, PUCA: A pseudonym scheme with user-controlled anonymity for vehicular ad-hoc networks (VANET), presented at *IEEE Vehicular Networking Conference (VNC)*, pp. 25-32, 2014.

[29] K. Zeng, Pseudonymous PKI for ubiquitous computing, presented at *European Public Key Infrastructure Workshop*, pp. 207-222, 2006.

[30] K. G. Paterson and G. Price, A comparison between traditional public key infrastructures and identity-based cryptography, in *Information Security Technical Report*, vol. 8, no. 3, pp. 57-72, 2003.

[31] J. Zhang and Y. Xu, Breaking and repairing of an anonymous and traceable communication protocol for vehicular ad hoc networks, presented at *IEEE 12th International Conference on Computer and Information Technology*, pp. 88-93, 2012.

[32] L. Chen, S. Ng, and G. Wang, Threshold anonymous announcement in VANETs, in *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 605-615, 2011.

[33] D. Derler, K. Samelin, D. Slamanig and C. Striecks, "Fine-Grained and Controlled Rewriting in Blockchains: Chameleon- Hashing Gone Attribute-Based", in *NDSS*, 2019.

[34] J. Ning, X. Huang, W. Susilo, K. Liang, X. Liu, Y. Zhang, "Dual Access Control for Cloud-Based Data Storage and Sharing", in *IEEE Transactions on Dependable and Secure Computing*, DOI: 10.1109/TDSC.2020.3011525.

[35] J. Ning, Z. Cao, X. Dong, K. Liang, L. Wei, K. R. Choo, "CryptCloud+: Secure and Expressive Data Access Control for Cloud Storage", in *IEEE Transactions on Services Computing*, vol. 29, no.1, pp. 111-124, 2021.

[36] J. Hur, D. Koo, Y. Shin, and K. Kang, Secure data deduplication with dynamic ownership management in cloud storage, in *IEEE Transactions on knowledge and data engineering*, vol. 28, no. 11, pp. 3113-3125, 2016.

[37] Pairing-Based Cryptography(PBC) library. [Online]. Available: https://crypto.stanford.edu/pbc/howto.html

[38] The GNU Multiple Precision Arithmetic Library (GMP). Accessed: Nov. 2017. [Online]. Available: http://gmplib.org

Huiying Hou received the B.S. and M.S. degrees from the College of Computer Science and Technology, Qingdao University, China, in 2015 and 2018, respectively. She is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Fudan University, China. Her research interests include applied cryptography and information security, in particular, vehicle ad-hoc network security and attribute-based cryptology.

Jianting Ning received the Ph.D. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University in 2016. He is currently a Professor with the Fujian Provincial Key Laboratory of Network Security and Cryptology, College of Mathematics and Computer Science, Fujian Normal University, China. Previously, he was a Research Scientist at School of Information Systems, Singapore Management University, and a Research Fellow at Department of Computer Science, National University of Singapore. His research interests include applied cryptography and information security, in particular, public key encryption, secure and privacy-preserving computation.

Yunlei Zhao received the Ph.D. degree in computer science from Fudan University, Shanghai, China, in 2004. He joined Hewlett-Packard European Research Center, Bristol, U.K., as a Post-Doctoral Researcher, in 2004. Since 2005, he has been with Fudan University, and is now a Distinguished Professor with School of Computer Science, Fudan University. His research interests include the theory and applications of cryptography.

Robert H. Deng is AXA Chair Professor of Cybersecurity and Director of the Secure Mobile Centre, School of Information Systems, Singapore Management University (SMU). His research interests are in the areas of data security and privacy, cloud security and Internet of Things security. He received the Outstanding University Researcher Award from National University of Singapore, Lee Kuan Yew Fellowship for Research Excellence from SMU, and Asia-Pacific Information Security Leadership Achievements Community Service Star from International Information Systems Security Certification Consortium. His professional contributions include an extensive list of positions in several industry and public services advisory boards, editorial boards and conference committees. These include the editorial boards of IEEE Security & Privacy Magazine, IEEE Transactions on Dependable and Secure Computing, IEEE Transactions on Information Forensics and Security, Journal of Computer Science and Technology, and Steering Committee Chair of the ACM Asia Conference on Computer and Communications Security. He is an IEEE Fellow.