Computer Science and Information Systems 15(3):549-567 https://doi.org/10.2298/CSIS171222022S

A Lightweight Batch Anonymous Authentication Scheme for VANET Based on Pairing-free

Cheng Song, Mingyue Zhang, Zongpu Jia, Weiping Peng, and Hairu Guo

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, Henan, 454003, China
songcheng@hpu.edu.cn, zmyue@njust.edu.cn,
{jiazp, pwp9999, guohr}@hpu.edu.cn

Abstract. Aimed at improving the security and efficiency of anonymous authentication in vehicular ad hoc network (VANET), a certificateless batch anonymous authentication scheme without bilinear pairings is put forward. By coordinating Trust Authority (TA) and vehicles to generate the public/private key pairs and pseudonyms, the system security is freed from dependency on tamper-proof devices. Through comprehensive analyses, this scheme is proved not only to be able to realize such security properties as authentication, anonymity, traceability, unforgeability, forward or backward security, etc., but also able to resist Type I and Type II attacks in the random oracle model. Moreover, this scheme effectively reduces system storage load by means of certificateless authentication, and the authentication efficiency can also be increased by realizing batch authentication based on pairing-free calculation. Accordingly, the scheme is proved to be significant in theory and valuable in application in the Internet of Things or embedded environment with limited resources.

Keywords: VANET; pairing-free; certificateless; anonymous authentication; random oracle model.

1. Introduction

Along with the wide use of vehicles in modern society, a series of transportation problems gradually emerge, like insufficiency of parking lots, traffic congestion, traffic accidents, etc. Consequently, such issues as traffic management, secure driving and traffic information exchange are drawing more and more attention. In managing large numbers of vehicles, Intelligent Transport System (ITS) [1] is widely adopted domestically and abroad. To establish the next-generation transport system, VANET [2]based on Mobile Ad hoc Network (MANET) [3] is emphasized more and more by enterprises and academia. In VANET, vehicles are able to obtain instant traffic information, weather information and entertainment information by communicating with roadside units (RSUs), which further enhances the enjoyment of driving and brings great convenience to vehicle users.

However, there are various kinds of threats in VANET environment: on the one hand, the data in wireless communications of VANET is easily detected, changed and faked; on the other hand, vehicles being in open physical environments, the leak of privacy (driver's identity, license plate number, position and itinerary) will cause some

threats to the lives and properties of both drivers and passengers. Consequently, user's privacy protection [4] is the fundamental demand for secure communications in VANET. Currently, the dominant method of privacy preservation is anonymous authentication [5], which, nevertheless, is complex in algorithm and relatively heavily-loaded in computation. Besides, the topological structure is caused to vary constantly by the wireless communications in VANET and the highly dynamic vehicles. As a result, the communication efficiency of information interaction in VANET draws more and more attention. Therefore, to improve the security and efficiency of anonymous authentication becomes one of the urgent tasks in VANET research. Aiming at some shortcomings, we proposed a lightweight batch anonymous authentication scheme for VANET. Compared with other schemes, we don't use the bilinear pairing algorithm with high computation complexity. So our scheme effectively reduces the computation cost. What's more, our scheme reduces the storage cost greatly because of certificateless.

The rest of this paper is organized as follows: In section 2, we introduce the related work. In Section 3, we introduce the VANET model. The proposed protocol is described in detail in Section 4. In Section 5, we give a theoretical proof of the scheme, such as correctness, anonymity, forward and backward privacy, and under the random oracle model we prove that the scheme can resist Type I and Type II attack, and an efficiency analysis is provided too. The last section concludes the paper.

2. Related Work

In recent years, researchers have conducted lots of researches on the privacy protection and data security for VANET and have put forward different anonymous authentication schemes. Raya, et al. firstly proposed an anonymous authentication protocol based on alias certificate [6], in which pseudonyms are adopted as user's true identity to protect user's private data. SPECS [7] introduced a batch verification protocol with enhanced security and privacy, in which a single vehicle could form a group with any other vehicles after batch verification and communicate with each other safely without involving RSU. However, Shi-Jinn, et al. [8] affirmed that SPECS couldn't resist impersonation attack, i.e. the masked vehicle could release false messages and even communicate safely with other legitimate vehicles. In Ref. [8], for ensuring privacy, each vehicle employs different pseudonyms generated by RSU when sending messages so that it needn't be equipped with large numbers of public/private key pairs, whereas the security of RSU couldn't be guaranteed. To solve this problem, Tzeng proposed an identity-based anonymous authentication scheme [9], in which each vehicle is equipped with a tamper-proofing device to store system's master key, the vehicle could generate pseudonyms by applying this master key. Nevertheless, recent study [10] shows that the security of Tzeng's scheme depends too much on the tamper-proofing device and attackers can obtain private information via side-channel attacks (like laser scanning and efficiency analysis). As a result, once tamper-proofing device is attacked and the master key is leaked the whole system's security will be threatened. Wang et al. proposed an effective identity privacy protection scheme ECPB [11] based on group signature. All members in the scheme need to be authenticated before they join the group, and also can realize batch authentication.

In the Ref. [12], an ID-based batch authentication scheme is proposed, where pseudonyms in each message signature process are assumed to be different, and the traceability can be achieved, too. Shao et al. proposed a threshold anonymous authentication protocol based on group signature which can achieve batch certification [13]. Lu et al. proposed a privacy protection scheme SPRING [14], which uses the constantly changing pseudonyms to protect the privacy of users. With the participation of trusted authority (TA), the efficiency of authentication is improved.

3. VANET Model

Different from the traditional Internet, VANET mainly adopts wireless communication mode, and the communication entity is vehicle. The system model includes three parties: trusted service center Trust Authority (TA), roadside unit RSU and vehicle unit OBU. The communications are two types: communication between vehicle and RSU and communication between vehicle and vehicle. The system network model is shown in Fig. 1.

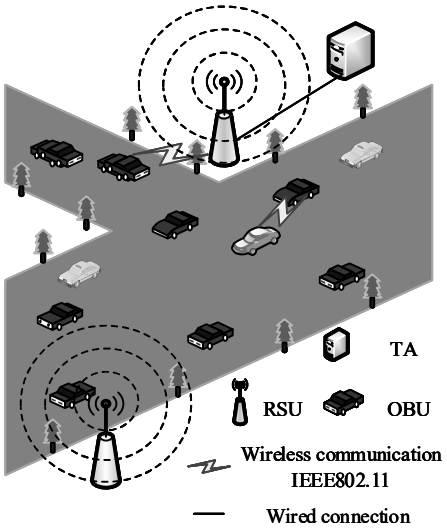

Fig. 1. VANET Model

Trusted Authority TA. In order to ensure the normally running of the system, TA is required to storing the privacy information for all the authenticated vehicles, generating the overall security parameters and distributing public/private keys to all participants. In general, vehicle manufacturer or transportation management department acts as TA.

Roadside unit RSU. Similar to the access nodes of wireless sensor networks, RSU is the infrastructure installed on both sides of the road, capable of communicating with

vehicles via wireless. The RSU communicates with the vehicle using the DSRC protocol [15], which enables RSU to validate the request information sent by the vehicle.

Vehicle unit OBU. In VANET, each vehicle is equipped with wireless communication module OBU, through which vehicles can communicate with RSU or other vehicles equipped with OBU.

4. The Proposed Scheme

The scheme proposed in this paper is composed of five phases: initialization phase, secret key generation phase, signature phase, authentication phase and update phase.

4.1. Initialization

l is a safety parameter of TA, TA randomly selects prime numbers p and q satisfies s $q|p-1$, then randomly selects $g \in Z_p^*$, $x \in Z_q^*$, computes $y = g^x \bmod p$, where $g^q = 1 \bmod p$, $g \neq 1$. The master key of the system is x, and the public key is $P_{pub} = y$. Define three $hash$ functions $H:\{0,1\}^* \to Z_q^*$, $H_1:\{0,1\}^* \to Z_q^*$, $H_2:\{0,1\}^* \to Z_q^*$. Then the system's public parameters are $\left(p, q, g, P_{pub}, H, H_1, H_2\right)$.

4.2. Secret Key Generation

Step1: Let ID_i be the identity of vehicle user Vi, Vi randomly selects $(b_i, v_i) \in Z_q^*$ as secret values, then computes:

$$B_i = g^{v_i} \bmod p \tag{1}$$

$$PID_{i_1} = b_i p \tag{2}$$

and transmits (ID_i, PID_{i_1}, B_i) to TA via secure channel.

Step 2: TA receives (ID_i, PID_{i_1}, B_i), randomly selects $r \in Z_q^*$, and calculates partial private key:

$$PS_i = r - xs_i \bmod q \tag{3}$$

then calculates

$$PID_{i_2} = ID_i \oplus H(xPID_{i_1} \| PID_{i_1} \| t_i \| P_{pub}) \tag{4}$$

The pseudonym of Vi is $PID_i = (PID_{i_1}, PID_{i_2}, t_i)$, where t_i is the valid time of PID_i.

$$S_i = H_1(PID_i \| B_i \| PP_i) \tag{5}$$

The partial public key is:

$$PP_i = g^r \bmod p \tag{6}$$

then transmits (PP_i, PS_i, PID_i) to Vi.

Step3: After Vi receives PP_i and PS_i sent by TA, it firstly verifies that $g^{PS_i} P_{pub}^{s_i} \stackrel{?}{=} PP_i \bmod p$, if holds, then goes to step 4.

Step4: Vi computes:

$$SK_i = v_i - PS_i \tag{7}$$

Let SK_i be the secret key, $PK_{i_1} = B_i$, $PK_{i_2} = PP_i$, and set the public key as $PK_i = (PK_{i_1}, PK_{i_2})$.

The flowchart in public/private-key-generating phase is shown in **Fig. 2**:

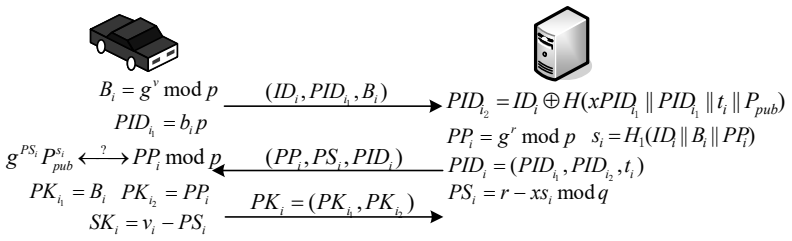

Fig.2. Message exchange flowchart in public/private-key-generating phase

4.3. Signature Phase

Vi randomly selects $k \in Z_q^*$, computes:

$$h_i = g^k \bmod p \tag{8}$$

$$u_i = H_2(PID_i \| m \| h_i \| tt_i) \tag{9}$$

$$e_i = k - SK_i u_i \bmod q \tag{10}$$

tt_i is the current timestamp, then let $\delta_i = (u_i, h_i, e_i, tt_i)$ be the signature of message m, transmit (δ_i, m, PID_i) to RSU.

4.4. Authentication Phase

In this section, the authentication is divided into two parts: single vehicle authentication and batch authentication.

Single vehicle authentication. Single vehicle authentication refers to a single vehicle request RSU or other vehicles for authentication. After the RSU or vehicle receives the message signature (δ_i, m, PID_i) from the vehicle Vi, they firstly check the validity of t_i and whether the timestamp tt_i is in a valid time interval, then computes:

$$S_i^{'} = H_1(PID_i \| PK_{i_2} \| PK_{i_2}) \tag{11}$$

$$u_i^{'} = H_2(PID_i \| m \| h_i \| tt_i) \tag{12}$$

then verifies that

$$(\frac{PK_{i_1} P_{pub}^{s_i^{'}}}{PK_{i_2}})^{u_i^{'}} \stackrel{?}{=} \frac{h_i}{g^{e_i}} \bmod p \tag{13}$$

If it holds, the authentication succeeds; otherwise, the message is rejected.

Batch authentication. Supposing that n different messages are (δ_1, m_1, PID_1), (δ_2, m_2, PID_2) ,… , (δ_n, m_n, PID_n) , where $\delta_i = (u_i, h_i, e_i)$, $PID_i = (PID_{i_1}, PID_{i_2}, t_i)$, $i = 1 \cdots n$.

According to the sources of the signature message, the batch authentication is divided into three types:

(1) Different messages from the same vehicle, i.e. all the PID_i are the same in each certification message.

(2) Same messages from different vehicles, i.e. all messages m are the same in each certification message.

(3) Different messages from different vehicles.

No matter what kind of type of the batch authentication, we can authenticate them via a common method.

When the RSU or the vehicle receives the batch authentication messages, the RSU or vehicle firstly checks the validity of t_i and whether the timestamps tt_i are in a valid time interval, then computes: $S_i^{'} = H_1(PID_i \| PK_{i_2} \| PK_{i_2})$ $u_i^{'} = H_2(PID_i \| m \| h_i \| tt_i)$ for each authentication message, and then verifies that

$$(P_{pub})^{\sum_{i=1}^{n} s_i^{'} u_i^{'}} \stackrel{?}{=} \prod_{i=1}^{n} (\frac{h_i}{g^{e_i}} (\frac{PK_{i_2}}{PK_{i_1}})^{u_i^{'}}) \bmod p \tag{14}$$

If it holds, it means that these distinct n signatures are valid.

For the first type, the n authentication messages are from the same vehicle, all the PID_i are the same, PK_{i_1} , PK_{i_2} and $s_i^{'}$ are respectively the same. So, the authentication formula can be simplified to:

$$(\frac{PK_{i_1} P_{pub}^{s_i^{'}}}{PK_{i_2}})^{\sum_{i=1}^{n} u_i^{'}} \stackrel{?}{=} \prod_{i=1}^{n} \frac{h_i}{g^{e_i}} \bmod p \tag{15}$$

The flowchart in signature authentication phase is shown in Fig. 3:

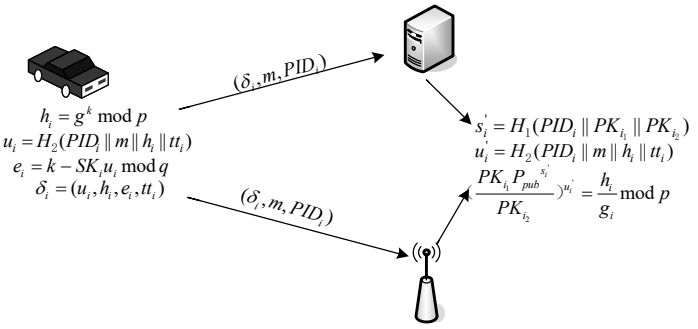

Fig. 3. Message exchange flowchart in signature authentication phase

4.5. Update Phase

In VANET, if the node is revoked or a new node is added, TA will provide real-time updates to the vehicles or RSU, inform them which PK_i is latest addition and which PK_i is invalid. So, in the verification phase, if another vehicle or RSU receives the authentication information sent by the cancelled vehicle Vi, they will not find the corresponding PK_i to these messages, thus, messages from the cancelled vehicle will be invalid. Moreover, there is an effective timestamp for PID_i, and even when the updates are not timely, the valid time of PID_i will also be the limiting factor for the verification.

5. Analysis

5.1. Correctness Analysis

This section is to prove the correctness of single authentication and the correctness of batch authentication, as is shown the following.

The correctness of single authentication. During single vehicle authentication, the single vehicle is valid or not according to the formula: $(\frac{PK_{i_1} P_{pub}^{s_i'}}{PK_{i_2}})^{u_i'} \stackrel{?}{=} \frac{h_i}{g^{e_i}} \bmod p$.

Because:

$$g^{e_i} PK_{i_1}^{u_i'} P_{pub}^{s_i' u_i'} = g^{e_i} PK_{i_1}^{u_i'} P_{pub}^{s_i' u_i'} PK_{i_2}^{-u_i'} PK_{i_2}^{u_i'}.$$

$$= g^{k - SK_i u_i \bmod q} PK_{i_1}^{u_i'} P_{pub}^{s_i' u_i'} (g^{PS_i} P_{pub}^{s_i} \bmod p)^{-u_i'} PK_{i_2}^{u_i'}$$

$$= g^{k-(v_i-(r-xs_i) \bmod p)u_i \bmod q} PK_{i_1}{}^{u_i'} P_{pub}{}^{s_i'u_i'} (g^{(r-xs_i) \bmod q} (g^x \bmod p)^{s_i})^{-u_i'} PK_{i_2}{}^{u_i'}$$

$$= g^{k-(v_i-(r-xs_i) \bmod p)u_i \bmod q} (g^{v_i} \bmod p)^{u_i'} (g^x \bmod p)^{s_i'u_i'} (g^{(r-xs_i) \bmod q} (g^x \bmod p)^{s_i})^{-u_i'} PK_{i_2}{}^{u_i'}$$

$$= g^{k-v_iu_i+ru_i-xs_iu_i} g^{v_iu_i'} g^{xs_i'u_i'} g^{(r-xs_i)-u_i'} g^{-xs_iu_i'} PK_{i_2}{}^{u_i'} \bmod p$$

$$= g^{k-v_iu_i+ru_i-xs_ih_i+v_iu_i'+xs_i'u_i'-u_i'r+xs_iu_i'-xs_iu_i'} PK_{i_2}{}^{u_i'} \bmod p$$

$$= g^k PK_{i_2}{}^{u_i'} \bmod p$$

$$= h_i PK_{i_2}{}^{u_i'} \bmod p .$$

$$g^{e_i} PK_{i_1}{}^{u_i'} P_{pub}{}^{s_i'u_i'} = h_i PK_{i_2}{}^{u_i'} \bmod p$$

$$(PK_{i_1} P_{pub}^{s_i'})^{u_i'} = \frac{h_i PK_{i_2}{}^{u_i'}}{g^{e_i}} \bmod p$$

$$\frac{(PK_{i_1} P_{pub}^{s_i'})^{u_i'}}{PK_{i_2}{}^{u_i'}} = \frac{h_i}{g^{e_i}} \bmod p$$

So, the single vehicle authentication is correct.

The correctness of batch authentication. During batch authentication, the batch vehicles are valid or not by verifying that

$$(P_{pub})^{\sum_{i=1}^{n} s_i'u_i'} \stackrel{?}{=} \prod_{i=1}^{n} (\frac{h_i}{g^{e_i}} (\frac{PK_{i_2}}{PK_{i_1}})^{u_i'}) \bmod p .$$

Because:

$$(\frac{PK_{1_1} P_{pub}{}^{s_1'}}{PK_{1_2}})^{u_1'} (\frac{PK_{2_1} P_{pub}{}^{s_2'}}{PK_{2_2}})^{u_2'} \cdots (\frac{PK_{i_1} P_{pub}{}^{s_i'}}{PK_{i_2}})^{u_i'} \cdots (\frac{PK_{n_1} P_{pub}{}^{s_n'}}{PK_{n_2}})^{u_n'}$$

$$= (\frac{h_1}{g^{e_1}} \frac{h_2}{g^{e_2}} \cdots \frac{h_i}{g^{e_i}} \cdots \frac{h_n}{g^{e_n}}) \bmod P$$

$$(\frac{PK_{1_1}}{PK_{1_2}})^{u_1'} (\frac{PK_{2_1}}{PK_{2_2}})^{u_2'} \cdots (\frac{PK_{i_1}}{PK_{i_2}})^{u_i'} \cdots (\frac{PK_{n_1}}{PK_{n_2}})^{u_n'} P_{pub}^{\sum_{i=1}^{n} s_i'u_i'} = \prod_{i=1}^{n} \frac{h_i}{g^{e_i}} \bmod p$$

$$\prod_{i=1}^{n} (\frac{PK_{i_1}}{PK_{i_2}})^{u_i'} P_{pub}^{\sum_{i=1}^{n} s_i'u_i'} = \prod_{i=1}^{n} \frac{h_i}{g^{e_i}} \bmod p$$

$$P_{pub}^{\sum_{i=1}^{n} s_i'u_i'} = \prod_{i=1}^{n} (\frac{h_i}{g^{e_i}} (\frac{PK_{i_2}}{PK_{i_1}})^{u_i'}) \bmod p$$

So, the batch authentication is correct.

5.2. Security Analysis

Privacy protection. In this scheme, each pseudonym PID_i involves the system master key x and b_i selected randomly by user, while x and b_i is only known to TA and Vi respectively. Based on discrete logarithm assumption, when x and b_i are unknown, it difficult for any attacker to compute PID_i. Consequently, even if the attacker obtains the pseudonym PID_i of Vi, it's unable to obtain any identity information of Vi.

Traceability. If a vehicle user sends an abnormal message, TA can still track the malicious vehicle even though it releases messages through a pseudonym. TA has system master key x. According to the pseudonym $PID_i = (PID_{i_1}, PID_{i_2}, t_i)$ and the equation $PID_{i_2} = ID_i \oplus H(xPID_{i_1} \| PID_{i_1} \| t_i \| P_{pub})$, TA can get $ID_i = PID_{i_2} \oplus H(xPID_{i_1} \| PID_{i_1} \| t_i \| P_{pub})$ then the true identity ID_i of Vi can be deduced. Consequently, when the signature is invalid, TA could trace vehicle's responsibility based on the signature.

Unlinkability. User's unlinkability means that attacker is unable to judge whether two messages are from the same vehicle or not. The unlinkability of the program is proved by the linking game. Set this scheme as η, challenger as A, signer RSU as ζ, while B_0 and B_1 stands for two loyal vehicle users.

Definition 1: Linking game

Step 1: A adopts key-generating algorithm $\mathrm{KeyGen}(k)$ to generate public/private key pairs (SK, PK), and obtains system's public parameters $\left(p, q, g, P_{pub}, H, H_1, H_2\right)$;

Step 2: A selects two different messages: m_0 and m_1;

Step 3 : Select random bit $b \in \{0,1\}$, then send m_b and m_{1-b} to B_0 and B_1 secretly, with b being a secrecy for the challenger;

Step 4: Signer ζ conducts this signature scheme respectively with B_0 and B_1;

Step 5: If B_0 and B_1 output two valid signatures δ_b and δ_{1-b} which respectively correspond to messages m_0 and m_1, then send δ_b and δ_{1-b} to challenger in random order; otherwise, return $\perp$ to the challenger;

Step 6: The challenger guesses that δ_b is from b', if $b' = b$, then A wins this game.

This paper defines that the advantage of A winning this game is: $Adv_{\eta, \mathrm{A}}^{Link}(A) = |2Pr[b' = b] - 1|$, where $Pr[b' = b]$ stands for the probability of $b' = b$.

Theorem 1. If any A fails to win this linking game with significant probability in polynomial time, then this scheme is proved to have unlinkability.

In the linking game, if the A receives $\perp$ in Step 5, then it means that A couldn't obtain any useful information, and the probability of obtaining the correct b is 1/2, which is equivalent to the random guess of b.

Suppose another situation: after implementing this signature scheme, A obtains two signatures (δ_0, m_0, PID_0) and (δ_1, m_1, PID_1). To prove the unlinkability of this scheme, let $j \in \{0,1\}$ and $(\delta_i, m_i, PID_i) \in \{(\delta_0, m_0, PID_0), (\delta_1, m_1, PID_1)\}$, if signature is guaranteed to be valid, $s_j^{'} = H_1(PID_j \| PK_{j_1} \| PK_{j_2})$ and $u_j^{'} = H_2(PID_j \| m_j \| h_j \| tt_j)$ can invariably be realized, then validate the equation $(\frac{PK_{j_1} P_{pub}^{s_j^{'}}}{PK_{j_2}})^{u_j^{'}} = \frac{h_j}{g^{e_j}} \bmod p$. In this way, challenger is unable to identify from which signer the message is sent, hence this scheme is possessed with unlinkability.

Forward and backward security. Forward security means that even if attacker obtains the currently-authenticated secret information, it will not be able to deduce the information related to the preceding authenticated message. Backward security means that even though attacker obtains the relevant information about authentication, it will not be able to deduce the subsequent authentication information so as to trace the authentication process of vehicles. In this scheme, if attacker obtains the signature message $\delta_i = (u_i, h_i, e_i, tt_i)$, where $h_i = g^k \bmod p$, $u_i = H_2(PID_i \| m \| h_i \| tt_i)$ and $e_i = k - SK_i u_i \mod q$, because k is randomly selected by Vi in every signature, consequently, attacker is impossible to deduce the preceding or subsequent $h_i^{'}$ based on the current h_i. Meanwhile, since u_i involves the randomly number k, pseudonym PID_i and signature message δ_i has validity period, and many random numbers are involved during computing e_i, so it is impossible for attacker to deduce the preceding or subsequent signature messages by means of current signature messages, thus this scheme satisfies forward and backward security.

Man-in-the-middle attack (MITM attack). In MITM attack, attacker keeps connected with two parties during communications, and makes them believe that they are communicating and exchanging information under safe circumstances, so as to acquire useful information to attack. In this scheme, a random number will be generated in each communication between RSU and Vi, whereas the random number used by attacker in establishing connection with RSU (or Vi) is different from the random number generated in the communications between RSU and Vi, therefore, attacker is unable to communicate with valid users to implement MITM attack by establishing communications connection with valid users.

Type I attack. Type I attack refers to exterior attacker which is able to replace valid users' public keys.

Theorem 2. Under the assumption of Discrete Logarithm Problem (DLP), the proposed scheme can implement the unforgeability of the adaptive selection message attack in the random oracle model.

Lemma 1. In Game I, within the limited time t, suppose Type I attacker AI is able to issue q_{par} queries to the partial private key extraction oracle, launch q_{pub} queries to public key extraction oracle, q_{pubr} queries to public key replacement oracle, q_{H_1} and

q_{H_2} queries to random oracle H_1 and H_2 respectively, and launch q_{sig} queries to signing oracle to output a valid signature with the probability ε, then there is an algorithm B, which can solve DLP with the probability:

$$\varepsilon' > (\varepsilon - \frac{1}{2l}) \times (1 - \frac{1}{q_{par}})^{q_{par}} \times (\frac{1}{2^{|p|}})^{q_{pubr}} \times (1 - \frac{1}{q_{par}})^{q_{pri}} \times \frac{1}{q_{par}},$$

where $|p|$ is the bit length of Z_p.

$$t < t' + (q_{pub} + 3q_{pubr} + 8q_{sig})t_e + (2q_{pub} + 3^{q_{pubr}} + 6q_{sig})t_m,$$

where t_m refers to the time needed in conducting a single modular multiplication, and t_e refers to the time needed in a single modular exponentiation.

In the following, it is proved that there exists an algorithm B can solve DLP with the help of AI in random oracle model:

Give B a random challenge tuple (p, g, β) of DLP and aims to output α satisfy $g^{\alpha} = \beta \bmod p$, algorithm B initializes AI with system parameter $(p, q, g, p_{pub}, H, H_1, H_2)$, then B respond the oracle queries from AI as a challenger. The specific process of queries in oracle model is as follows:

Partial private key extraction queries: When AI quires this oracle model with PID_i, B records the answers between AI and B with the list $L_{par} = (PID_i, PS_i)$. If B finds (PID_i, PS_i) in the list L_{par}, then B returns PS_i to AI; otherwise, B randomly selects $c \in [1, q_{par}]$.

(1) If $i \neq c$, B randomly selects $PS_i \in Z_q^*$, transmits PS_i to AI, and stores (PID_i, PS_i) in the list L_{par}.

(2) If $i = c$, B validates $PID_i = PID^*$, outputs "failure" and halts.

Public key extraction queries: When AI queries this oracle model by inputting PID_i, and records the answers between AI and B with $L_{pub} = (PID_i, PK_{i_1}, PK_{i_2}, s_i, v_i)$. If B finds $(PID_i, PK_{i_1}, PK_{i_2}, s_i, v_i)$ in L_{pub}, then B transmits (PK_{i_1}, PK_{i_2}) to AI; otherwise:

(1) If $PID_i = PID^*$, B randomly selects $PK_{i_2} \in Z_p^*$ to calculate $s_i \in Z_p^*$, sets $PK_{i_1} = PK_{i_2} \beta^{-1} P_{pub}^{-s_i} \bmod p$, transmits (PK_{i_1}, PK_{i_2}) to AI, and stores $(PID_i, PK_{i_1}, PK_{i_2}, s_i, \perp)$ in L_{pub} ..

(2) If $PID_i \neq PID^*$, B recovers (PID_i, PS_i) from L_{par}; if B fails to find (PID_i, PS_i) in the list L_{par}, it obtains a new PS_i by performing partial private key extraction queries. Then B randomly selects $s_i \in Z_p^*$ and $v_i \in Z_p^*$, calculates

$PK_{i_2} = g^{ps_i} P_{pub}^{s_i} \bmod p$ and $PK_{i_1} = g^{v_i} \bmod p$, then B transmits (PK_{i_1}, PK_{i_2}) to AI, and stores (PID_i, PS_i) and $(PID_i, PK_{i_1}, PK_{i_2}, s_i, v_i)$ in L_{par} and L_{pub}.

Public key replacement queries: When AI queries this oracle with the tuple, $(PID_i, \widetilde{PK_{i_1}}, \widetilde{PK_{i_2}})$, B recovers (PID_i, PS_i) from L_{par}; if B fails to find (PID_i, PS_i) in L_{par}, it obtains a new PS_i by performing partial private key extraction queries, then B verifies whether the equation $g^{PS_i} P_{pub}^{H_1(PID_i \| \widetilde{PK_{i_1}} \| \widetilde{PK_{i_2}})} = PK_{i_2} \bmod p$ is valid or not. If the equation is valid, B outputs "failure" and stops operating; otherwise, B continues to carry out the following two steps:

(1) If B finds $(PID_i, PK_{i_1}, PK_{i_2}, s_i, v_i)$ in L_{pub}, B validates $PK_{i_1} = \widetilde{PK_{i_1}}$ and $PK_{i_2} = \widetilde{PK_{i_2}}$, then stores $(PID_i, \widetilde{PK_{i_1}}, \widetilde{PK_{i_2}}, \perp, \perp)$ in L_{pub}.

(2) Otherwise, B obtains a new PK_{i_1} and PK_{i_2} by performing queries of public key extraction with the input PID_i, then validates and ,, and stores $(PID_i, \widetilde{PK_{i_1}}, \widetilde{PK_{i_2}}, \perp, \perp)$ in L_{pub}.

H_1 **queries**: When AI conducts H_1 queries by inputting (PID_i, B_i, PP_i), assume that AI has already obtained B_i and PP_i by public key extraction queries, that is, PK_{i_1} and PK_{i_2}. Consequently, B finds the corresponding $(PID_i, PK_{i_1}, PK_{i_2}, s_i, v_i)$ in L_{pub}, and returns s_i to AI.

H_2 **queries**: When AI conducts H_2 queries by inputting (PID_i, m_i, h_i, tt_i), B records the queries between AI and B wit L_2 in the form of $(PID_i, m_i, u_i, h_i, tt_i)$. If B finds $(PID_i, m_i, u_i, h_i, tt_i)$ in L_2, then B returns u_i to AI; otherwise B randomly selects $u_i \in Z_q^*$, returns u_i to AI, and stores $(PID_i, m_i, u_i, h_i, tt_i)$ in L_2.

Private key extraction queries: When AII quires this oracle model by inputting PID_i:

(1) If $PID_i = PID^*$, B will output "failure" and stop operating.

(2) If $PID_i \neq PID^*$, B recovers (PID_i, PS_i) in the list L_{par} and recovers $(PID_i, PK_{i_1}, PK_{i_2}, s_i, v_i)$ from L_{pub}; if B fails to find (PID_i, PS_i) and $(PID_i, PK_{i_1}, PK_{i_2}, s_i, v_i)$ from the corresponding list, it obtains new PS_i and e_i by issues a partial private key extraction query and a public key extraction query on PID_i. Then B transmits $v_i - PS_i$ to AII, and stores $(PID_i, PK_{i_1}, PK_{i_2}, s_i, v_i)$ and (PID_i, PS_i) in the lists L_{pub} and L_{par}.

Signing queries: AI conducts H_2 queries by inputting (PID_i, m_i), assuming PID_i has been queried before.

(1) If $PID_i \neq PID^*$, B outputs signature δ_i corresponding to message m_i, and transmits δ_i to AI;

(2) Otherwise, B randomly selects $e_i, u_i \in Z_q^*$, recovers the corresponding $(PID_i, PK_{i_1}, PK_{i_2}, s_i, v_i)$ in L_{pub}, and calculates $h_i = g^{e_i} PK_{i_1}^{u_i} P_{pub}^{s_i u_i} PK_{i_2}^{-u_i}$. $\delta_i = (u_i, h_i, e_i, tt_i)$ serves as a valid signature sent from signer PID_i to message m_i. B returns δ_i to AI and stores $(PID_i, m_i, u_i, h_i, tt_i)$ in L_2. If (PID_i, m_i, h_i, tt_i) has already been recorded in L_2 or PK_{i_1} and PK_{i_2} have been replaced, B will output "failure" and halts.

Forgery: Finally, AI stops queries and outputs a valid signature $(\hat{h}, \hat{u}, \hat{e})$ with respect $(\widehat{PID}, \widehat{m})$. If $\widehat{PID} \neq PID^*$, then B outputs "failure" and halts; otherwise, B repeats the operation in the similar random way, while the choices in H_2 queries are different. B can obtain the other valid signature $(\hat{h}, \hat{u}', \hat{e}')$, both signatures should satisfy:

$g^{\hat{e}} PK_i^{\hat{u}} P_{pub}^{s\hat{u}} = \hat{h} PK_i^{\hat{u}}$ and $g^{\hat{e}'} PK_i^{\hat{u}'} P_{pub}^{s\hat{u}'} = \hat{h} PK_i^{\hat{u}'}$

Consequently, B satisfies $g^{\hat{e}-\hat{e}'} = \beta^{\hat{u}-\hat{u}'} \Leftrightarrow \log_g \beta = \dfrac{\hat{e}-\hat{e}'}{\hat{u}-\hat{u}'}$, then judged from $\dfrac{\hat{e}-\hat{e}'}{\hat{u}-\hat{u}'}$, B can successfully solves DLP.

Probability analysis: We analyze the probability of B winning the game. Since H_2 is a random queries oracle, the probability of AI generating a valid signature corresponding to $(\widehat{PID}, \widehat{m})$ without queries $H_2(\widehat{PID} \| \hat{m} \| \hat{h})$ is at least $1/2^l$. The probability of B not halts in public key replacement simulation is $(1-1/2^{|p|})^{q_{pubr}}$; the probability of B not stopping simulation in private key extraction queries is $(1-1/q_{par})^{q_{pri}}$; the probability of B continuing normal operation in DLP computation phase is $1/q_{par}$. Therefore, the probability of B winning the game is at least:

$$(\varepsilon - \frac{1}{2l}) \times (1 - \frac{1}{q_{par}})^{q_{par}} \times (\frac{1}{2^{|p|}})^{q_{pubr}} \times (1 - \frac{1}{q_{par}})^{q_{pri}} \times \frac{1}{q_{par}}$$

the maximum time needed in B operation is:

$$t + (q_{pub} + 3q_{pubr} + 8q_{sig})t_e + (2q_{pub} + 3^{q_{pubr}} + 6q_{sig})t_m$$

Type II attack. Type II attacker refers to internal attacker which is able to obtain the system master key.

Lemma 2. In Game II, within the limited time t', suppose Type II attacker AII is able to launch q_{par} queries to partial private key extraction oracle, launch q_{pub} queries to public key extraction oracle, launch q_{H_1} and q_{H_2} queries to random oracle models H_1 and H_2 respectively, launch q_{pri} queries to private key extraction oracle, and launch q_{sig} queries to signing oracle to output a valid signature: $t' < t + (3q_{pub} + 8q_{sig})t_e + (3q_{pub} + 6q_{sig})t_m$ with the probability ε, then there is an algorithm B ,which can solve DLP with the probability: $\varepsilon' > (\varepsilon - \frac{1}{2l}) \times (1 - \frac{1}{q_{pub}})^{q_{pri}} \times \frac{1}{q_{par}}$. Where t_m refers to the time needed in executing a single modular multiplication, and t_e refers to the time needed in a single modular exponentiation.

In the following, it is proved that an algorithm B exists to solve DLP with the help of AII. The specific proof process is as follows:

Give B a random challenge tuple (p, g, β) of DLP, aims to output α satisfy $g^{\alpha} = \beta \bmod p$, algorithm B initializes AI with system parameter $(p, q, g, p_{pub}, H, H_1, H_2)$ and the system master key x, then B undertakes the queries from AII as a challenger. The specific process of queries in oracle model is as follows:

Public key extraction queries: When AII queries this oracle model by inputting PID_i, challenger B records the answers between AII and B with $L_{pub} = (PID_i, PK_{i_1}, PK_{i_2}, PS_i, s_i, v_i)$. If B finds $(PID_i, PK_{i_1}, PK_{i_2}, PS_i, s_i, v_i)$ in the list L_{pub}, then B transmits (PK_{i_1}, PK_{i_2}) to AII; otherwise, B randomly selects $c \in [1, q_{par}]$:

(1) If $i \neq c$, B randomly selects $PS_i \in Z_q^*$, $s_i \in Z_p^*$, $v_i \in Z_p^*$ to calculate $PK_{i_2} = g^{ps_i} P_{pub}^{s_i} \bmod p$, $PK_{i_1} = g^{v_i} \bmod p$, transmits (PK_{i_1}, PK_{i_2}) to AII, and stores $(PID_i, PK_{i_1}, PK_{i_2}, PS_i, s_i, v_i)$ in the list L_{pub}.

(2) If $i = c$, B validates $PID_i = PID^*$, randomly selects $PS_i \in Z_q^*$, $s_i \in Z_p^*$, $v_i \in Z_p^*$ to calculate $PK_{i_2} = g^{ps_i} P_{pub}^{s_i} \bmod p$, $PK_{i_1} = PK_{i_2} \beta^{-1} P_{pub}^{-s_i} \bmod p$, transmits (PK_{i_1}, PK_{i_2}) to AII, and stores $(PID_i, PK_{i_1}, PK_{i_2}, PS_i, s_i, v_i)$ in the list L_{pub}.

H_1 **queries**: When AII queries H_1 with inputting (PID_i, B_i, PP_i), assume that AII has made public key extraction queries on PID_i to obtain B_i and PP_i, which are actually

PK_{i_1} and PK_{i_2} . Consequently, B finds the corresponding $(PID_i, PK_{i_1}, PK_{i_2}, PS_i, s_i, v_i)$ in L_{pub} , and returns s_i to AII.

H_2 **queries:** When AII conducts H_2 queries by inputting (PID_i, m_i, h_i, tt_i) , B records the answers between AII and B with L_2 in the form of $(PID_i, m_i, u_i, h_i, tt_i)$. If B finds $(PID_i, m_i, u_i, h_i, tt_i)$ in L_2 , then B returns u_i to AII; otherwise B randomly selects $u_i \in Z_q^*$, returns u_i to AII, and stores $(PID_i, m_i, u_i, h_i, tt_i)$ in L_2 .

Private key extraction queries: When AII queries this oracle by inputting PID_i :

(1) If $PID_i = PID^*$, B will output "failure" and stop operating.

(2) If $PID_i \neq PID^*$, B recovers $(PID_i, PK_{i_1}, PK_{i_2}, PS_i, s_i, v_i)$ from L_{pub} ; if B fails to find $(PID_i, PK_{i_1}, PK_{i_2}, PS_i, s_i, v_i)$ from L_{pub} , it obtains new PS_i and e_i by issuing public key extraction query. Then B transmits $v_i - PS_i$ to AII, and stores $(PID_i, PK_{i_1}, PK_{i_2}, PS_i, s_i, v_i)$ in L_{pub} .

Signing queries: AII conducts H_2 queries by inputting (PID_i, m_i) , assuming PID_i has been queries before.

(1) If $PID_i \neq PID^*$, B outputs signature δ_i corresponding to message m_i , and transmits δ_i to AII;

(2) Otherwise, B randomly selects $e_i, u_i \in Z_q^*$, recovers the corresponding $(PID_i, PK_{i_1}, PK_{i_2}, PS_i, s_i, v_i)$ in L_{pub} , and calculates $h_i = g^{e_i} PK_{i_1}^{u_i} P_{pub}^{s_i u_i} PK_{ii_2}^{-u_i} \bmod p$. $\delta_i = (u_i, h_i, e_i, tt_i)$ serves as a valid signature sent from signer PID_i to message m_i . B returns δ_i to AII and stores $(PID_i, m_i, u_i, h_i, tt_i)$ in L_2 . If (PID_i, m_i, h_i, tt_i) has already been recorded in L_2 , B will output "failure" and stop simulating .

Forgery and Probability analysis: After all the queries, B can obtain $\frac{\hat{e} - \hat{e}'}{\hat{u} - \hat{u}'}$ by a similar approach to Lemma 1, then B can solve the DLP problem. And we use the same way as Lemma 1 to get the probability of B winning the game is at least:

$$(\varepsilon - \frac{1}{2l}) \times (1 - \frac{1}{q_{par}})^{q_{pri}} \times \frac{1}{q_{par}}$$

the maximum time needed in B operation is:

$$t + (3q_{pub} + 8q_{sig})t_e + (3q_{pub} + 6q_{sig})t_m$$

Comparison of security properties in the proposed scheme with that of other schemes is shown in Table 1. The proposed scheme can further enhance the security of anonymity authentication for VANET.

Table 1. Comparison of security

schemes	Ref.[11]	Ref.[12]	Ref.[13]	Ref.[14]	Ref.[16]	the proposed scheme
anonymity	√	√	√	√	√	√
traceability	√	√	√	√	√	√
unlinkability			√		√	√
for/backward security	√				√	√
MITM attack		√		√	√	√
Type Ⅰ						√
Type Ⅱ						√

5.3. Efficiency Analysis

Computation complexity. For the sake of qualitative analysis and comparison in analysis of computation complexity, we use T_{mul} to denote a point multiplication operation in elliptic curve, T_{par} to denote a bilinear pairing operation, T_{exp} to denote an exponential operation, and T_h to denote a MapToPoint $hash$ function operation. In the experiment of the Ref.[16], runs 2 GHz CPU, 4-GB RAM processor to find that the operation time of T_{exp} is 0.6ms, T_{par} is 1.6ms, T_h is 2.7ms and T_{mul} is 0.6ms. The comparison with the existing schemes is shown in Table 2:

Table 2. Comparison of computation costs

schemes	authenticate one message	authenticate n messages
Ref.[11]	$5T_{\text{par}} + 12T_{\text{exp}}$	$2T_{par} + 13nT_{\text{exp}}$
Ref.[12]	$T_{mul} + 2T_{par} + T_h$	$nT_{mul} + 2T_{par} + nT_h$
Ref.[13]	$10T_{par} + 4T_{\text{exp}}$	$(n+6)T_{par} + 4nT_{\text{exp}}$
Ref.[14]	$11T_{mul} + 3T_{par}$	$11nT_{mul} + 3nT_{par}$
Ref.[16]	$2T_{\text{par}} + 5T_{\text{exp}}$	$(n+1)T_{\text{par}} + (4+n)T_{\text{exp}}$
The proposed scheme	$6T_{\text{exp}}$	$(2\text{n} + 4)T_{\text{exp}}$

As is shown in the table, the proposed scheme doesn't require bilinear pairings operation which is quite complicated in computation; instead, it adopts exponential operation with low computation complexity. Therefore, this scheme has obvious advantages in computational complexity concerning either single message authentication or batch authentication.

Communication complexity. Communication complexity means the number of bytes needed in communications. The communication overhead of a single complete verification in VANET authentication scheme is mainly composed of identity information, signatures and messages, etc.

Let the length of original message be 20 bytes. In scheme [11], the signature of message is 220 bytes, timestamp is 4 bytes, and ID is 2 bytes. In scheme [12], original message is 20 bytes, signature is 60 bytes, pseudonym is 40 bytes, timestamp is 4 bytes, and ID space is 4 bytes. In scheme [13], the original message is 20 bytes., signature space is 826 bytes, timestamp is 4 bytes, and ID space is 3 bytes. In scheme [14], signature is 40 bytes, certificate is 121 bytes, anonymous secret key is 26 bytes, and ID space is 2 bytes. In scheme [16], the original message is 20 bytes, signature is 20 bytes, the public key is 20 bytes, anonymous the certificate is 180 bytes. In this proposed scheme, original message is 20 bytes, signature is 60 bytes, and pseudonym is 41 bytes, as is shown in **Table 3**:

Table 3. Comparison of communication complexity (unit: bytes)

schemes	single message authentication
Ref.[11]	$220+4-2=226$
Ref.[12]	$20+60+40+4+4=128$
Ref.[13]	$20+826+4+3=853$
Ref.[14]	$40+121+26+2=189$
Ref.[16]	$20+20+20+180=240$
the proposed scheme	$20+60+41=121$

The comparison demonstrates that this proposed scheme has certain advantage in communication complexity.

6. Conclusions

To solve the efficiency problem in anonymity authentication in existing privacy protection schemes for VANET, this paper proposes a certificateless batch anonymity authentication scheme without bilinear pairings operation. Analyses shows that this scheme is not only guaranteed many security properties like anonymity, traceability, unlinkability, forward and backward security, etc. on the basis of correctness. We prove that this scheme can resist Type I and Type II attack under random oracle model. Meanwhile, the public/private keys in this scheme are distributed generated, which effectively avoids the dependence on tamper-resistant device inside vehicle. Furthermore, by adopting certificateless batch authentication without bilinear pairings operation, computation and storage costs are cut down considerably. Therefore, this scheme has much theoretical significance and applied value in the resource-limited Internet of Things or embedded environment.

Despite the proposed scheme has made improvement and optimization on the basis of the existing schemes, with the rapid development of vehicle networking technology, it

is difficult for anonymous authentication schemes under traditional VANET models to meet the requirements of privacy protection in vehicle anonymous authentication in various communication environments. Therefore, it remains to be tackled for the author to further improve and optimize the proposed scheme so as to satisfy the growing demands of communication of VANET in various conditions.

Acknowledgement. This work was supported by the Natural Science Foundation of China project (61300124, 61300216, 61872126); Henan Provincial Department of Science and Technology Research Project (132102210123).

References

1. Giovanna C., Giuseppe, M., Antonio, P., et al.: Transport models and intelligent transportation system to support urban evacuation planning process[J]. Iet Intelligent Transport Systems, vol.10, no.4, pp. 279-286. (2016)
2. Rizvi, M., Pasha, S., Tamrakar, S.: MANET Parameter Analysis and its Impact on Next Generation Network[C]// International Conference on Recent Trends in Computer Science and Electronics Engineering. (2017)
3. Chouhan, P., Kaushal, G., Prajapat, U.: Comparative Study MANET and VANET[J]. International Journal of Advanced Trends in Computer Science & Engineering. (2016)
4. Diep, P T N., Yeo, C K.: A trust-privacy framework in vehicular ad hoc networks (VANET)[C] //Wireless Telecommunications Symposium (WTS), 2016. IEEE, pp. 1-7. (2016)
5. Liu, F B., Zhang, K., Li, H., et al.: Threshold traceability anonymous authentication scheme without trusted center for Ad Hoc network[J]. Journal on Communications, vol.33, no.8, pp.208-213. (2012)
6. Raya, M., Hubaux, J P.: Securing vehicular ad hoc networks[J]. Journal of Computer Security, vol.15, no.1, pp.39-68. (2007)
7. Chim, T W., Yiu, S M., Hui, L C K., et al.: SPECS: Secure and privacy enhancing communications schemes for VANETs[J]. Ad Hoc Networks, vol.9, no.2, pp.189-203. (2011)
8. Horng, S J., Tzeng, S F., Pan, Y., et al.: b-SPECS+: Batch Verification for Secure Pseudonymous Authentication in VANET[J]. IEEE Transactions on Information Forensics & Security, vol.8, no.11, pp.1860-1875. (2013)
9. Horng, S J., Tzeng, S F., Li, T., et al.: Enhancing Security and Privacy for Identity-based Batch Verification Scheme in VANET[J]. pp.1-1. (2017)
10. Mahanta, H J., Azad, A K., Khan, A K.: Differential Power Analysis: Attacks and Resisting Techniques[M]// Information Systems Design and Intelligent Applications. Springer India, pp.349-358. (2015)
11. Wang, Y., Zhong, H., Xu, Y., et al.: ECPB: Efficient conditional privacy-preserving authentication scheme supporting batch verification for VANETs[J]. International Journal of Network Security, vol.18, no.2, pp.374-382. (2016)
12. Shim, K A.: Reconstruction of a Secure Authentication Scheme for Vehicular Ad Hoc Networks Using a Binary Authentication Tree[J]. IEEE Transactions on Wireless Communications, vol.12, no.11, pp. 5386-5393. (2013)
13. Shao, J., Lin, X., Lu, R., et al.: A Threshold Anonymous Authentication Protocol for VANETs[J]. IEEE Transactions on Vehicular Technology, vol.65, no.3, pp.1-1. (2016)
14. Lu, R., Lin, X., Shen, X.: SPRING: A Social-based Privacy-preserving Packet Forwarding Protocol for Vehicular Delay Tolerant Networks[C]// Conference on Information Communications. IEEE Press, pp. 632-640. (2010)

15. Tong, Z., Lu, H., Haenggi, M., et al.: A Stochastic Geometry Approach to the Modeling of DSRC for Vehicular Safety Communication[J]. IEEE Transactions on Intelligent Transportation Systems, vol.17, no.5, pp.1-11. (2016)
16. Azees, M., Vijayakumar, P., Deboarh, L J.: EAAP: Efficient Anonymous Authentication With Conditional Privacy-Preserving Scheme for Vehicular Ad Hoc Networks[J]. IEEE Transactions on Intelligent Transportation Systems, vol.1, no.10, pp(99).1-10. (2017)

Cheng Song was born in 1980 in Guangshan City, Henan Province. He received the Ph.D. degree from the Beijing University of Posts and Telecommunications in 2011. He is now an lecture at the School of Computer Science and Technology in Henan Polytechnic University. His main research interests include Networking Security and Application, Privacy protection and Trusted Computing.

Mingyue Zhang, the corresponding author of this paper, received the master's degree from Henan Polytechnic University in July 2018. she is currently a Ph.D. student at Nanjing University of Science and Technology. Her research focuses on network information security and privacy protection

Zongpu Jia was born in 1963 in Dengzhou City, Henan Province. He received the Ph.D. degree from Jilin University. He is now a professor at the School of Computer Science and Technology in Henan Polytechnic University and a doctoral supervisor. His main research interests are in the areas of Computer Network Technology, Internet of Things Technology and Application, and Information Systems. He has published more than 80 articles over a series of research topics.

Weiping Peng was born in 1979 in Tianmen, Hubei, China. He received the Ph.D. degrees from Beijing University of Posts and Telecommunications in 2011.He is now an associate professor in the School of Computer Sciences and Technology, Henan Polytechnic University. His research interests include Data Loss Protection,information security and Trusted Computing.

Hairu Guo was born in 1977 in Shuozhou City, Shanxi Province. He received the Ph.D. degree from the Beijing University of Posts and Telecommunications in 2013. He is now an associate professor at the School of Computer Science and Technology in Henan Polytechnic University. His main research interests include Artificial Intelligence, Quantum Computing.

Received: December 22, 2017; Accepted: August 2, 2018.