Computer Modeling in Engineering & Sciences
DOI:10.32604/cmes.2020.013289
Article

An Anonymous Authentication Scheme with Controllable Linkability for Vehicle Sensor Networks

Yousheng Zhou[1,2], Lvjun Chen[1], Xiaofeng Zhao[1,*] and Zheng Yang[3]

[1]College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China
[2]School of Cyber Scurity and Information Law, Chongqing University of Posts and Telecommunications, Chongqing, 400065, China
[3]Singapore University of Technology and Design, 8 Somapah Rd, 487372, Singapore
*Corresponding Author: Xiaofeng Zhao. Email: 1466532412@qq.com
Received: 01 August 2020; Accepted: 30 September 2020

Abstract: Vehicle sensor networks (VSN) play an increasingly important part in smart city, due to the interconnectivity of the infrastructure. However similar to other wireless communications, vehicle sensor networks are susceptible to a broad range of attacks. In addition to ensuring security for both data-at-rest and data-in-transit, it is essential to preserve the privacy of data and users in vehicle sensor networks. Many existing authentication schemes for vehicle sensor networks are generally not designed to also preserve the privacy between the user and service provider (e.g., mining user data to provide personalized services without infringing on user privacy). Controllable linkability can be used to facilitate an involved entity with the right linking key to determine whether two messages were generated by the same sender, while preserving the anonymity of the signer. Such a functionality is very useful to provide personalized services. Thus, in this paper, a threshold authentication scheme with anonymity and controllable linkability for vehicle sensor networks is constructed, and its security is analyzed under the random oracle model.

Keywords: Threshold authentication; controllable linkabilty; group signature; vehicle sensor networks

1 Introduction

While vehicle sensor networks research is fairly mature [1], there is plenty of research opportunities in this space due to continuing and rapid advances in vehicular communication technology and other underpinning technologies (e.g., smart/driverless vehicles and other Internet-connected technologies in a smart city). In vehicle sensor networks, there are two key types of entities—see Fig. 1, namely: wireless on-board units (OBUs) on vehicles to supply wireless communication ability, and roadside unit (RSU) located on the road or buildings within a certain coverage. Normally, a remote central authority (CA) is also deployed to assist OBUs or RSU to perform a given task, such as authentication. These parties can support two types of communications, namely: Vehicle-to-infrastructure (V2I) communication and vehicle-to-vehicle

(V2V) communication [2,3]. Such communications can be used to support activities such as reporting of traffic congestion and accidents/incidents. However, due to characteristics such as self-organizing, rapid-changing and open channel, vehicle sensor networks are susceptible to a broad range of attacks. Achieving secure and efficient authentication services is a basic and critical component [4–6], but increasingly there are other properties/features that should be considered. Examples include privacy preservation [7–9], and the related notions such as anonymity and unlinkability [10–12].

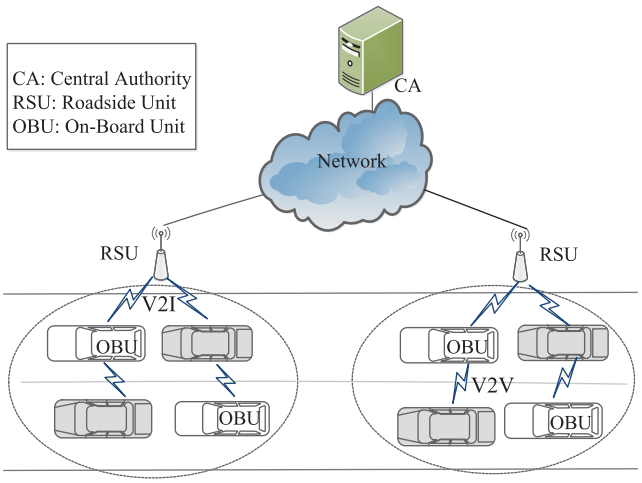

Figure 1: Entities of vehicle sensor network

In general, striking a balance between preserving user privacy and maximizing the utility of user data (e.g., to offer better and customized services, based on mining and analysis of user data) is tricky [13,14]. For example, a key characteristic required to provide personalized services is linkability, which contradicts the privacy requirement. Controllable linkability, first proposed by Hwang et al. [15], is one potential solution. In such a concept, an entity who owns a linking key can dervie whether two authentication messages were generated by the same user (or not). Doing so does not infringe the user's anonymity since the identity of the message signer cannot be obtained. Since the seminal work of Hwang et al. [15], a great many group signature schemes with controllable linkability have been investigated in the literature [15–18]. However, the verifier can only check the valid signature message generated by a group member but cannot decide whether the message has been fabricated. Threshold authentication can, however, mitigate such a limitation. Specifically, the receiver accepts a message only after it has been confirmed by the specified threshold number of user.

In this work, we present a group signature-based anonymous authentication scheme for vehicle sensor networks, which is designed to achieve threshold authentication, anonymity, non-repudiation, and controllable linkability. In addition, we will demonstrate that it is more efficient than similar existing schemes in regard to both communicational and computational costs, based on the findings from our evaluations using the widely accepted OpenSSL library. We also demonstrate the security of the scheme under the random oracle model, as well as explaining how it achieves the other desirable security properties.

The rest of this article is structured as below. Related work and relevant background materials are introduced in Sections 2 and 3, respectively. Then, the concrete construction of the scheme is presented in Section 4, followed by its security and performance analysis in Sections 5 and 6. Finally, this paper is concluded in the last section.

2 Related Work

In recent years, authentication schemes with different properties for vehicle sensor networks have been investigated in the literature. For instance, Raya et al. [19] introduced an anonymous authentication scheme for vehicle sensor networks by employing anonymous certificates. In such a scheme, a vehicle is preloaded with large anonymous certificates such that the vehicle can employ different public/private key pairs during each authentication process to avoid being traced. However, the public/private key pairs must have a short lifetime so as to achieve privacy preservation; otherwise, there will be significant storage and management costs. Lu et al. [20] presented a new method to deal with the challenge of preloading a mass of anonymous certificates, by leveraging RSUs. To update the anonymous certificate in order to keep linkability of the message, each vehicle would request the RSU to issue a short-time anonymous certificate when the vehicle passes by the RSU. Consequently, frequent interaction between vehicle and RSU may influence the performance of the entire vehicle sensor networks. Huang et al. [21] proposed two certificateless signatures schemes; however, anonymity is not achieved because the public key of the user is needed during verification.

Group signature schemes can also be used to achieve privacy preservation [22–24]. For example, Hwang et al. [15–17] introduced three group signature schemes with controllability linkability, for purpose of preserving the privacy between the users and service providers. However, these schemes do not support threshold authentication and require significant computing cost due to the number of exponentiation operations and bilinear pairings operations.

Threshold authentication is a common approach to assure the authenticity of the received (traffic) information [25–27]. For example, Shao et al. [28,29] introduced two threshold anonymous authentication schemes for vehicle sensor networks, designed to resist an attack on a single malicious message. However, the cost of computation of these schemes is significantly high on account of the employment of exponentiation and bilinear pairing.

Therefore, in this work, we construct a group signature-based anonymous authentication scheme with controllable linkability, based on Shao et al. [28,29] scheme. However, our proposed scheme is more efficient because we utilize the point multiplication operation instead of the exponentiation operations.

3 Preliminaries

3.1 System and Security Models

Our proposed protocol comprises four entities, namely: central authority (CA), service providers (SP), RSUs and OBUs (see also Fig. 2). CA is mainly tasked with issuing of the corresponding public key certificates for both RSUs and OBUs after their respective public keys have been successfully authenticated. Moreover, CA can uncover the original identity of the sender who is found to send a fabricated message in VANET. SP is responsible for providing personalized services, first by examining whether given two messages are produced by the same sender with the linking key. RSUs are densely deployed along the road, and each of them is assumed as the manager of a group consisting of OBUs within its communication area. Besides, RSUs are also responsible for issuing group certificates for vehicles equipped with OBUs when

they enter into its communication range, which can be used to communicate with other OBUs by signing the message with its private key. Note that if an OBU is in the revocation list obtained from the CA, it would not be assigned with a group certificate by its RSU.

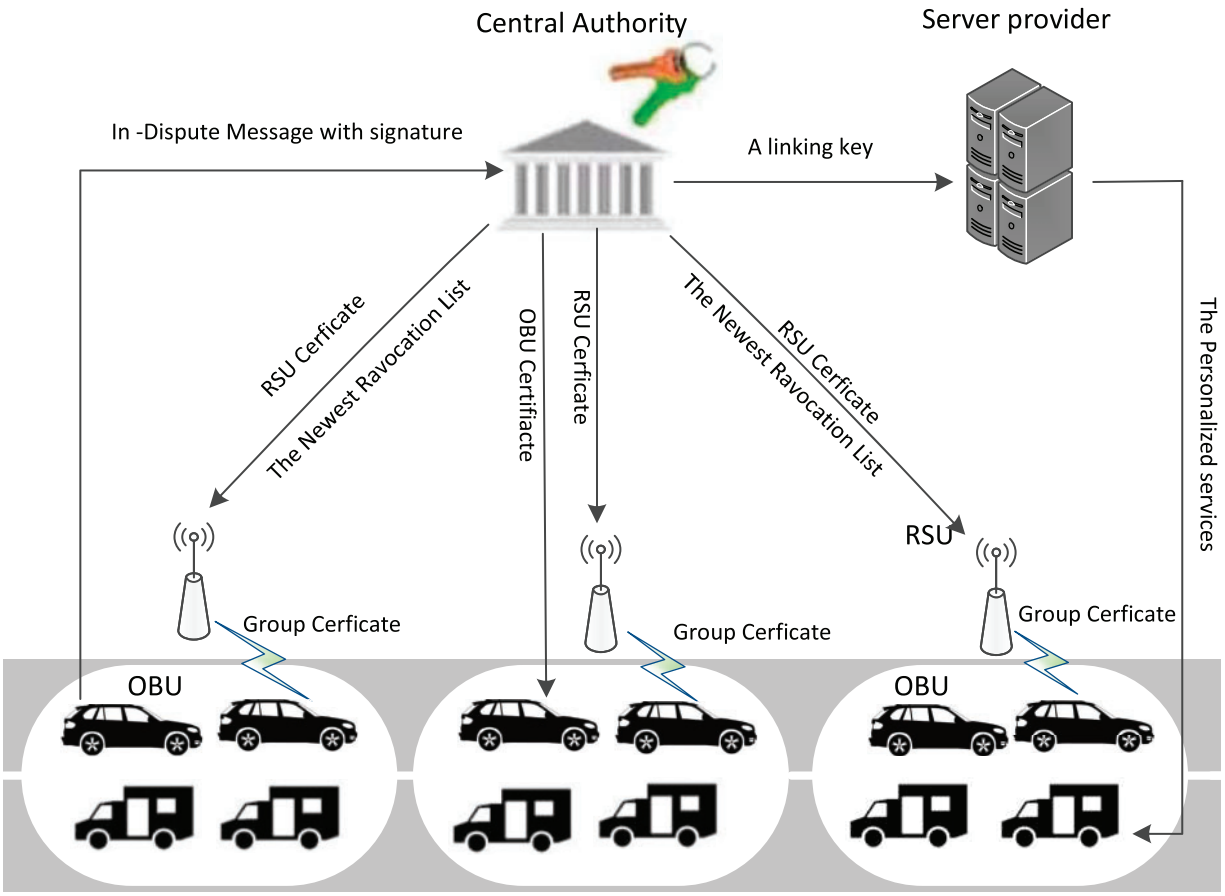

Figure 2: System model

CA is assumed to be fully honest, whereas SPs and RSUs are presumed to be semi-honest (i.e., honest but curious), in the sense that they would honestly follow the proposed protocol and would not conspire with other RSUs. However, they are curious about the user's identity information and trace information, and hence may passively seek to collect group signatures and gather other information. Honest OBUs can accept a message only when they have received the number of valid signatures whose number is greater than the threshold value on the same message. However, OBUs could also be malicious, in the sense of attempting to obtain the user's identity information and trace information by launching either passive or active attack. For instance, they may attempt to broadcast many fabricated message signatures without being perceived or conspire with each other.

3.2 Bilinear Groups

Let G_1, G_2 and G_3 denote three different additive groups over elliptic curve with the same order q, where q is a prime number, and they all satisfy non-degenerated properties and are used

to construct a bilinear map $e: G_1 \times G_2 \to G_3$, such that $e(aP_1, b\widetilde{P_1}) = e(P_1, \widetilde{P_1})^{ab}$ for all $a, b \in Z_q^*$, any $P_1 \in G_1$ and $\widetilde{P_1} \in G_2$. For convenience, the symbol "$\sim$" is used to label the elements in G_2.

We analyze the security of the proposed threshold anonymous authentication scheme based on the eCDH assumption and the eDDH assumption, which are defined as follows [29]:

Definition 1: (eCDH Assumption): Given $P, aP, bP \in G_1$ and $\widetilde{P}, a\widetilde{P} \in G_2$, where $a, b \in Z_q^*$, to output abP. The (t, ε) eCDH assumption states that there is no t-time algorithm that can break the eCDH assumption with a non-negligible advantage of at least ε.

Definition 2: (eDDH Assumption): Given $P, aP, bP, cP \in G_1$ and $\widetilde{P}, a\widetilde{P}, b\widetilde{P} \in G_2$, where $a, b, c \in Z_q^*$, to decide whether $abP = cP$ holds or not. The (t, ε) eDDH assumption states that there is no t-time algorithm can break the eDDH assumption with non-negligible advantage of at least ε.

4 Proposed Authentication Protocol

The construction of our proposed group signature-based anonymous authentication scheme with controllable linkability is illustrated here, and the scheme includes initialization, registration, joining, signing, verifying, linking, and tracing stage.

First, the CA follows the initialization process to produce public/private key pairs for itself and the public parameters for the entire system. Before each RSU and OBU join the network, they need to follow the registration process to produce the pairs of the public key and private key for itself and obtain corresponding public certificates from the CA. RSUs are deployed on critical points along the road (e.g., roadsides or building and other installations). When a vehicle employed with an OBU enters into a new range covered by a certain RSU, it has to follow the joining process to obtain the corresponding group certificate from the RSU. Then, the vehicle can sign and broadcast messages. After that, the receiver can perform the threshold authentication process to verify any received messages and signatures. In order to identify the malicious signer, the CA can perform identity tracing process to uncover the identity of the singer corresponding to the suspicious signature. To provide personalized service, one can perform linking process to check whether two given pairs of signatures and messages are from the same sender.

The definition of used notations is shown as Tab. 1, and details of our proposed authentication scheme is illustrated in the remaining of this section.

4.1 Initialization

In this stage, CA produces the key pairs for itself and the public parameters for the entire system. The detailed description is as follows:

- First, CA produces the public parameter $q, P_1, P_2 \in G_1, \widetilde{P}_1 \in G_2, e: G_1 \times G_2 \to G_3, H_1(\cdot): \{0,1\}^* \to G_1, H_2(\cdot): \{0,1\}^* \to Z_q^*$.
- Then, CA randomly chooses $x_{ca}, x_{tm} \in Z_q^*$, and computes $P_{ca} = x_{ca}P_1, \widetilde{P}_{ca} = x_{ca}\widetilde{P}_1$ and $\widetilde{P}_{tm} = x_{tm}\widetilde{P}_1, P_{link} = -x_{tm}P_1$. Finally, CA sets P_{link} as the linking key, $(P_{ca}, \widetilde{P}_{ca}, \widetilde{P}_{tm})$ as its public key and keeps (x_{ca}, x_{tm}) as its private key.

4.2 Registration

The registration stage consists of two parts, namely: RSU registration and OBU registration. CA assign RSUs and OBUs with the corresponding public certificates by performing this process.

Table 1: Summary of notations

Notation	Definitions
q	A secure large prime
G_1, G_2, G_3	Three different groups with the same order q
P_1, P_2	The primitive generator of G_1
$\widetilde{P}_1$	The primitive generator of G_2
x_{ca}	The private key of CA to issue certificates
x_{tm}	The private key of CA to trace
x_{rsu}	The private key of RSU
x_{obu}	The private key of OBU
P_{link}	The linking key of SP
$(P_{ca}, \widetilde{P}_{ca}, \widetilde{P}_{tm})$	The public key of CA
$\widetilde{P}_{rsu}$	The public key of RSU
$\widetilde{P}_{obu}$	The public key of OBU
Z_q^*	The collection including all primes in $\{0, 1, \ldots, q-1\}$
H_1	A hash function mapping to G_1
H_2	A hash function mapping to Z_q^*
τ	A signature of message

4.2.1 RSU Registration

Each RSU registers itself as follows,

- RSU selects $x_{rsu} \in Z_q^*$ randomly as its private key, and evaluates $\widetilde{P}_{rsu} = x_{rsu}\widetilde{P}_1$ as its public key.
- RSU sends $\widetilde{P}_{rsu}$ to CA through a secure channel. After receiving the message, CA produces a public certificate $cert_{rsu}$ on $\widetilde{P}_{rsu}$, and sends $cert_{rsu}$ and the current revocation list CRL to RSU, where CRL is defined as

$$CRL = ((cert_{obu\_1}, \widetilde{P}'_{obu\_1}), (cert_{obu\_2}, \widetilde{P}'_{obu\_2}), \ldots, (cert_{obu\_n}, \widetilde{P}'_{obu\_n}))$$

4.2.2 Vehicle OBU registration

- Each OBU selects $x_{obu} \in Z_q^*$ randomly as its private key and evaluates $P_{obu} = x_{obu}P_1$ as its public key.
- Then, OBU sends P_{obu} and $\widetilde{P}_{obu} = x_{obu}\widetilde{P}_1$ to CA through a secure channel. After receiving the message, if $e(P_{obu}, \widetilde{P}_1) = e(P_1, \widetilde{P}_{obu})$ holds, then CA produces corresponding public certificate $cert_{obu}$ on P_{obu}, and sends $cert_{obu}$ to the OBU. Finally, CA records $(cert_{obu}, \widetilde{P}_{obu})$ in the user list.

4.3 Joining

In this stage, RSUs will issue corresponding group certificate for the OBUs within their radio coverage. When OBU_i gets into the communication area covered by a new RSU, the joining stage is activated between OBU_i and the particular RSU. The detailed steps are as follows.

- To begin with, OBU_i sends a request message to RSU for obtaining its public key.
- Upon receiving the request from OBU_i, RSU returns its certificate and public key $(cert_{rsu}, \widetilde{P}_{rsu})$ to OBU_i.
- Upon receiving $(cert_{rsu}, \widetilde{P}_{rsu})$, OBU_i checks $(cert_{rsu}, \widetilde{P}_{rsu})$. If it is not valid, OBU_i would be required to send another request message again; otherwise, OBU_i selects $k, n \in Z_q^*$ randomly and computes $P'_{obu} = x_{obu} P_{ca}$. Then, it uses the public key of RSU $\widetilde{P}_{rsu}$ to encrypt P'_{obu}, where the encrypting process is found by computing $k\widetilde{P}_{rsu} = (x_1, y_1)$ and $C_{obu} = (k\widetilde{P}_1, P'_{obu} + x_1 P_1)$. Finally, OBU_i sends $(cert_{obu}, P_{obu}, C_{obu}, n)$ to RSU, where n is a random number chosen from Z_q^*.
- Upon receiving $(cert_{obu}, P_{obu}, C_{obu}, n)$, RSU uses its private key x_{rsu} to decrypt C_{obu} and obtains P'_{obu}, and checks whether $cert_{obu}$ exists in the revocation list CRL. Then it checks whether $e(P_{obu}, \widetilde{P}_{ca}) = e(P'_{obu}, \widetilde{P}_1)$. If it does not holds, then it terminates at this stage; otherwise, RSU chooses two random numbers $r, t \in Z_q^*$ and computes group certificate $cert_g = (c_1, c_2)$, where $c_1 = x_{rsu} P_2 - r(P'_{obu}), c_2 = rP_1$. Finally, RSU adds OBU_i's certificate $cert_{obu}$ to member list(ML) and uses OBU_i's public key P_{obu} to encrypt $cert_g$, where the encrypting process is found by computing $tP_{obu} = (x_2, y_2)$ and $C_{rsu} = (tP_1, c_2 + x_2 P_1, c_1 + x_2 P_1)$. It then broadcasts (C_{rsu}, n, CRL_{rsu}) within its communication range, where CRL_{rsu} is the latest and is obtained from CRL and $cert_{obu}$ exists in ML of this RSU.
- When OBU_i receives (C_{rsu}, n, CRL_{rsu}), OBU_i first determines whether this message is sent to itself by using the value n. If it holds, then OBU_i uses its private key x_{obu} to decrypt C_{rsu} and obtains $cert_g$, prior to checking whether $e(c_1, \widetilde{P}_1) \cdot e(x_{obu} c_2, \widetilde{P}_{ca}) = e(P_2, \widetilde{P}_{rsu})$. If it holds, then OBU_i accepts this group certificate $cert_g = (c_1, c_2)$; otherwise, OBU_i sends the request message to RSU again.

4.4 Signing

When an OBU intends to broadcast a message m, it performs the following steps to sign the message.

- OBU_i chooses $r', \alpha, s \in Z_q^*$ randomly.
- Randomizes the group certificate as $\tau_1 = c_1 - r'(x_{obu} P_{ca})$ and $\tau_2 = c_2 + r' P_1$.
- Encrypts $\widetilde{P}_{obu}$ for tracing as $\widetilde{\tau}_3 = \alpha \cdot \widetilde{P}_1$, $\widetilde{\tau}_4 = x_{obu} \cdot \widetilde{P}_1 + \alpha \cdot \widetilde{P}_{tm}$.
- Binds (τ_1, τ_2) and $\widetilde{\tau}_3, \widetilde{\tau}_4$ together by $\tau_5 = x_{obu} \cdot \tau_2$ and $\tau_6 = \alpha \cdot \tau_2$.
- Computes $\tau_7 = x_{obu} H_1(m)$, which would be employed to determine whether two given signatures for a certain message are produced by a same OBU or not. However, the characteristic of threshold authentication is enabled by τ_7.
- A bundle of the above evaluated values is made by $S_1 = s \cdot \tau_2$, $S_2 = s \cdot H_1(m)$, $\sigma_8 = H_2(m||\tau_1||\cdots||\tau_7||S_1||S_2)$, $\tau_9 = s - \tau_8 x_{obu}$.
- Set $\tau = \{\tau_1, \tau_2, \widetilde{\tau}_3, \widetilde{\tau}_4, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9\}$ and broadcast (m, τ).

4.5 Verifying

Upon receiving a message m and its signature τ, OBU_j uses CA's public key $(\widetilde{P}_{ca}, \widetilde{P}_{tm})$, RSU's public key $\widetilde{P}_{rsu}$, and the revocation list CRL_{rsu} to verify this signature as follows.

- Signature verification: Initially check if the signature $\{\tau_1, \tau_2, \widetilde{\tau}_3, \widetilde{\tau}_4, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9\}$ is valid by checking the following equations:

—$e(\tau_1, \widetilde{P}_1) \cdot e(\tau_5, \widetilde{P}_{ca}) = e(P_2, \widetilde{P}_{rsu})$
—$e(\tau_2, \widetilde{\tau}_3 + \widetilde{\tau}_4) = e(\tau_5, \widetilde{P}_1) \cdot e(\tau_6, \widetilde{P}_{tm} + \widetilde{P}_1)$
—$S_1 = \tau_9 \tau_2 + \tau_8 \tau_5$
—$S_2 = \tau_9 H_1(m) + \tau_8 \cdot \tau_7$
—check $\tau_8 = H_2(m||\tau_1|| \cdots ||\tau_7||S_1||S_2)$

- Revocation check: Check whether the signer within this RSU range is not revoked, by checking the equation $e(\tau_1, \widetilde{P}_1) \cdot e(\tau_2, \widetilde{P}'_{obu\_i}) \neq e(P_2, \widetilde{P}_{rsu})$, for all $\widetilde{P}'_{obu\_i} \in CRL_{rsu}$.

If all equations hold, then OBU_j believes the validity of the signature, i.e., the sender of the signature has not been revoked. Once OBU_j had received exceeding threshold number of valid signatures about the same message from distinctive OBUs, it would accept and believe the message.

In addition, the OBU can also use batch verification to speed up the verification on $\{m_1, \tau^1\}$, $\{m_2, \tau^2\}, \ldots, \{m_n, \tau^n\}$, as follows

- $e\left(\sum_{i=1}^{n} \tau_1^i, \widetilde{P}_1\right) \cdot e\left(\sum_{i=1}^{n} \tau_5^i, \widetilde{P}_{ca}\right) = e(P_2, \widetilde{P}_{rsu})$
- $\prod_{i=1}^{n} e\left(\tau_2^i, \widetilde{\tau}_3^i + \widetilde{\tau}_4^i\right) = e\left(\sum_{i=1}^{n} \tau_5^i, \widetilde{P}_1\right) \cdot e\left(\sum_{i=1}^{n} \tau_6^i, \widetilde{P}_{tm} + \widetilde{P}_1\right)$

4.6 Linking

With the linking key P_{link}, SP can check whether two given pairs (m', τ') and (m, τ) are generated by a same user, as follows.

- First, it performs the verification process to check the validity of two given signatures.
- If the pairs are not valid, $\perp$ would be returned; otherwise, it examines whether the equation $e(P_{link}, \widetilde{\tau}'_3) \cdot e(P_1, \widetilde{\tau}'_4) = e(P_{link}, \widetilde{\tau}''_3) \cdot e(P_1, \widetilde{\tau}''_4)$ holds or not. If yes, 1 would be returned, i.e., the pairs are linked; otherwise, 0 would be returned, i.e., the pairs are unlinked.

4.7 Tracing

In this stage, CA can recover the real identity of the sender corresponding to a valid pair (m, τ), then it updates the *CRL* and sends *CRL* to each RSU. The detailed process is as follows:

- First, CA reveals the identity of signer corresponding to the signature message (m, τ) by computing $\widetilde{P}_{obu} = \widetilde{\tau}_4 - x_{tm}\widetilde{\tau}_3$.
- Then, CA finds signer's certificate $cert_{obu}$ in user list and computes $\widetilde{P}'_{obu} = x_{ca}\widetilde{P}_{obu}$.
- Finally, CA records $(cert_{obu}, \widetilde{P}'_{obu})$ in *CRL* and sends *CRL* to each RSU.

5 Correctness and Security Analysis

In this section, the analysis of correctness and security about our proposed threshold anonymous authentication scheme are provided.

5.1 Correctness

We will now illustrate that our proposed scheme satisfies the correctness requirements according to Theorem 1.

Theorem 1: Our presented threshold anonymous authentication scheme is reasonable, i.e., the signatures generated by the honest user can be efficiently verified and traced correctly.

Proof: The signatures generated by the honest users can be efficiently verified as follows,

$$\begin{aligned}
e(\tau_1, \widetilde{P}_1) \cdot e(\tau_5, \widetilde{P}_{ca}) &= e(x_{rsu}P_2 - (r+r')(x_{ca}P_{obu}), \widetilde{P}_1) \\
&\quad \cdot e((r+r')P_{obu}, x_{ca}\widetilde{P}_1) \\
&= e(x_{rsu}P_2, \widetilde{P}_1) \cdot e(-(r+r')(x_{ca}P_{obu}), \widetilde{P}_1) \\
&\quad \cdot e((r+r')(P_{obu}), x_{ca}\widetilde{P}_1) \\
&= e(P_2, x_{rsu}\widetilde{P}_1) \cdot e(-(r+r')(x_{ca}P_{obu}), \widetilde{P}_1) \\
&\quad \cdot e((r+r')(x_{ca}P_{obu}), \widetilde{P}_1) \\
&= e(P_2, \widetilde{P}_{rsu})
\end{aligned}$$

$$\begin{aligned}
e(\tau_2, \widetilde{\tau}_3 + \widetilde{\tau}_4) &= e(\tau_2, \alpha\widetilde{P}_1 + x_{obu}\widetilde{P}_1 + \alpha\widetilde{P}_{tm}) \\
&= e(\tau_2, \alpha\widetilde{P}_1) \cdot e(\tau_2, x_{obu}\widetilde{P}_1) \cdot e(\tau_2, \alpha\widetilde{P}_{tm}) \\
&= e(\alpha\tau_2, \widetilde{P}_1) \cdot e(x_{obu}\tau_2, \widetilde{P}_1) \cdot e(\alpha\tau_2, \widetilde{P}_{tm}) \\
&= e(\tau_6, \widetilde{P}_1) \cdot e(\tau_5, \widetilde{P}_1) \cdot e(\tau_6, \widetilde{P}_{tm}) \\
&= e(\tau_5, \widetilde{P}_1) \cdot e(\tau_6, \widetilde{P}_{tm} + \widetilde{P}_1) \\
s \cdot \tau_2 &= (\tau_9 + \tau_8 x_{obu})\tau_2 = \tau_9\tau_2 + \tau_8\tau_5 \\
s \cdot H_1(m) &= (\tau_9 + \tau_8 x_{obu})H_1(m) = \tau_9 H_1(m) + \tau_8\tau_7
\end{aligned}$$

The signatures generated by the honest users can be traced correctly using the following equations:

$$\begin{aligned}
\widetilde{P}_{obu} &= \widetilde{\tau}_4 - x_{tm}\widetilde{\tau}_3 \\
&= x_{obu} \cdot \widetilde{P}_1 + \alpha \cdot \widetilde{P}_{tm} - \alpha \cdot x_{tm}\widetilde{P}_1 \\
&= x_{obu} \cdot \widetilde{P}_1
\end{aligned}$$

5.2 Security Analysis

We will now prove that our scheme achieves unforgeability and anonymity under the random oracle model, respectively in Theorems 2 and 3.

Unforgeability: In order to show that our anonymous authentication scheme satisfies unforgeability, we will prove that the adversary $\mathscr{A}$ cannot produce a valid signature in case it does not know secret key x_{obu} or group certificate $cert_g = (c_1, c_2)$. This security feature can be achieved by the unforgeability of signature.

Theorem 2: Our presented threshold anonymous authentication scheme is unforgeability.

Proof: We will demonstrate that if the unforgeability of our proposed authentication scheme can be violated by an adversary $\mathscr{A}$ with advantage ε, then an algorithm $\mathscr{B}$ can be built to break some hard problem by invoking $\mathscr{A}$ in a blackbox manner. Therefore, there exist two cases for the unforgeability of our proposed anonymous authentication scheme, the one is that the private key of group member is known to $\mathscr{B}$, but the corresponding group certificate is unknown, and

the other is that the group certificate is known to $\mathscr{B}$, but the private key of group member is unknown.

Case 1: In terms of the former case, the purpose of $\mathscr{B}$ is to solve the eCDH problem, i.e., with given $(P, aP, bP, \widetilde{P}, a\widetilde{P})$, $\mathscr{B}$ outputs abP. Firstly, $\mathscr{B}$ generates the public parameters. It sets $P_1 = aP$, $\widetilde{P}_1 = a\widetilde{P}$, $P_2 = bP$, $P_{ca} = w(aP) = wP_1$, $\widetilde{P}_{ca} = w(a\widetilde{P}) = w\widetilde{P}_1$, where $w, x_{tm} \in Z_q^*$ is selected randomly. $\mathscr{B}$ can interact with $\mathscr{A}$ by issuing the following queries.

- OBU public key oracle: When the adversary $\mathscr{A}$ issues this query to ask for the public key of OBUs, $\mathscr{B}$ returns $P_{obu} = x_{obu}P_1$ to $\mathscr{A}$, where x_{obu} is selected randomly from Z_q^*.
- RSU public key oracle: When the adversary $\mathscr{A}$ issues this query to ask for the public key of RSUs, $\mathscr{B}$ decides whether the public key is the target public key. $\mathscr{B}$ computes $\widetilde{P}_{rsu} = a\widetilde{P} = x_{rsu}\widetilde{P}_1$ if the public key is the target public key; otherwise, $\widetilde{P}_{rsu} = x_{rsu}\widetilde{P}_1$, where x_{rsu} is selected randomly from Z_q^*. At last, $\mathscr{B}$ returns $\widetilde{P}_{rsu}$ to $\mathscr{A}$.
- OBU private key oracle: When $\mathscr{A}$ requests the corresponding private key of the public key P_{obu} obtained from public key oracle, the corresponding x_{obu} is returned by $\mathscr{B}$.
- Signature generation oracle: When $\mathscr{A}$ asks for the signature with a message m and an OBU public key P_{obu} generated in OBU public key oracle and a RSU public key $\widetilde{P}_{rsu}$ generated in RSU public key oracle. If $\widetilde{P}_{rsu}$ is $a\widetilde{P}$, $\mathscr{B}$ reports failure; otherwise, B performs as follows. $\mathscr{B}$ firstly issues private key oracle to obtain x_{obu}, and then selects $r, \alpha, \tau_8, \tau_9 \in Z_q^*$ randomly. Finally, $\mathscr{B}$ computes the signature and returns the signature to $\mathscr{A}$.

  $\tau_1 = x_{rsu}P_2 - r \cdot (x_{obu}P_{ca}), \tau_2 = rP_1, \widetilde{\tau}_3 = \alpha \cdot \widetilde{P}_1,$

  $\widetilde{\tau}_4 = x_{obu}\widetilde{P}_1 + \alpha \cdot \widetilde{P}_{tm}, \tau_5 = x_{obu}\tau_2, \tau_6 = \alpha \cdot \tau_2,$

  $\tau_7 = x_{obu}H_1(m), S_1 = \tau_9\tau_2 + \tau_8\tau_5,$

  $S_2 = \tau_9 H_1(m) + \tau_8 \cdot \tau_7$
- Reveal oracle: Take a pair (m, τ) from $\mathscr{A}$ as input, $\mathscr{B}$ performs as the actual execution, as it knows x_{tm}.

In a moment, $\mathscr{A}$ outputs a valid signature $\{\tau_1, \tau_2, \widetilde{\tau}_3, \widetilde{\tau}_4, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9\}$ on the message m under the targeted public key of RSU $\widetilde{P}_{rsu} = a\widetilde{P}$. Whereafter, $\mathscr{B}$ is able to compute $abP = \tau_1 + w\tau_5 = abP - r(wP_{obu}) + w(rP_{obu})$, which exactly is the solution of the eCDH problem about the instance P, aP, bP. Note that only if $\mathscr{B}$ can correctly guess the right public key of RSU, the eCDH problem can be resolved with the probability of at least ε/qk_{rsu}, where qk_{rsu} denotes the number of RSU public key.

Case 2: For the later case, the purpose of $\mathscr{B}$ is still to solve the eCDH problem, i.e., with given $(P_1, aP_1, bP_1, \widetilde{P}_1, a\widetilde{P}_1)$, $\mathscr{B}$ outputs abP.

Firstly $\mathscr{B}$ generates public parameter and public and private key pair for CA and RSU as the actual execution. Then, $\mathscr{B}$ can interact with $\mathscr{A}$ by issuing the following queries.

- OBU public key oracle: When the adversary $\mathscr{A}$ issues this query to request the public key of OBUs, $\mathscr{B}$ returns $P_{obu} = t_{obu}(aP_1)$, where t_{obu} is selected randomly from Z_q^*.
- Group certificate oracle: When $\mathscr{A}$ requests the corresponding group certificate for the public key P_{obu} obtained from the public key oracle, $\mathscr{B}$ performs as the actual execution, because it knows the group certificate, so it knows x_{rsu}.
- H_1 hash oracle: When $\mathscr{A}$ inputs message m, $\mathscr{B}$ firstly checks whether (m_1, r_1, R_1) exists in List L_{h_1}. If it exists, R_1 is returned to $\mathscr{A}$; otherwise, $R_1 = r_1(bP_1)$ if the message m is

targeted message, where $r_1 \in Z_q^*$ is chosen randomly; $R_1 = r_1 P_1$, otherwise. Finally, the new tuple (m_1, r_1, R_1) is recorded into L_{h_1}.

- H_2 hash oracle: When $\mathscr{A}$ inputs message $m\|\tau_1\|\cdots\|\tau_7\|S_1\|S_2$, $\mathscr{B}$ firstly checks whether $(m\|\tau_1\|\cdots\|\tau_7\|S_1\|S_2, r_2)$ exists in List L_{h_2}. If it exists, r_2 is returned to $\mathscr{A}$; otherwise, a random element $r_2 \in Z_q^*$ is chosen and returned. Finally, the new tuple $(m\|\tau_1\|\cdots\|\tau_7\|S_1\|S_2, r_2)$ is recorded into L_{h_2}.
- Signature oracle: When $\mathscr{A}$ asks for this oracle with the message m and a public key P_{obu} obtained from the OBU public key oracle. $\mathscr{B}$ firstly selects $r, \alpha, \beta, \tau_8, \tau_9 \in Z_q^*$ randomly, then calculates and outputs the signature $\{\tau_1, \tau_2, \widetilde{\tau}_3, \widetilde{\tau}_4, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9\}$.

$$\tau_1 = x_{rsu}P_2 - r\cdot(x_{ca}P_{obu}), \quad \tau_2 = rP_1, \quad \widetilde{\tau}_3 = \alpha\cdot\widetilde{P}_1,$$

$$\widetilde{\tau}_4 = t_{obu}\cdot(a\widetilde{P}_1) + \alpha\cdot\widetilde{P}_{tm}, \quad \tau_5 = r\cdot t_{obu}\cdot(aP_1),$$

$$\tau_6 = \alpha\cdot\sigma_2, \quad \tau_7 = r_1\cdot t_{obu}\cdot(aP_1), \quad S_1 = \tau_9\tau_2 + \tau_8\tau_5,$$

$$S_2 = \tau_9 H_1(m) + \tau_8\cdot\tau_7$$

where t_{obu} is the value corresponding to P_{obu} and r_1 is the value corresponding to the message m in L_{h_1}. At last, $\mathscr{B}$ checks whether $(m\|\tau_1\|\cdots\|\tau_7\|S_1\|S_2, \sigma_8)$ exists in List L_{h_2}. If it does not exists, $\mathscr{B}$ records $(m\|\tau_1\|\cdots\|\tau_7\|S_1\|S_2, \sigma_8)$ into L_{h_2}; otherwise, aborts.

In a moment, $\mathscr{A}$ outputs a valid signature $\{\tau_1, \tau_2, \widetilde{\tau}_3, \widetilde{\tau}_4, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9\}$ about the target message m^* under P_{obu}^*. $\mathscr{B}$ can compute $abP = \left(\frac{1}{r^* t_{obu}^*}\right)\tau_7$, which exactly is the solution of the eCDH on the instance P, aP, bP, where t_{obu}^*, r^* are the corresponding values of P_{obu}^* and m^* in L_{h_1}.

Note that only if $\mathscr{B}$ correctly guessed the target message, the eCDH problem can be solved with the probability of at least $\frac{\varepsilon}{q_{H_1}}$, where q_{H_1} is the number of H_1 hash oracle.

Anonymous: In signature $\{\tau_1, \tau_2, \widetilde{\tau}_3, \widetilde{\tau}_4, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9\}$, the identity information of the signer is only included in $\tau_1, \widetilde{\tau}_4, \tau_5, \tau_7$. However, the identity included in $\widetilde{\tau}_4$ is encrypted using CA's public key, and the identity information in τ_1, τ_5, τ_7 can be check only via pairing operation because it is located in the exponent. However, τ_1, τ_5, τ_7 belong to G_1, there is no public value contained in the identity information in G_2.

Theorem 3 (Anonymity) our proposed signature scheme is anonymous.

Proof: We will demonstrate that if there exists an ε-advantage adversary can break the identity indistinguishability of the proposed scheme, then a polynomial probability time algorithm $\mathscr{B}$ can be built to solve the eDDH problem with an advantage at least ε, i.e., with given $(P, \widetilde{P}, aP, bP, a\widetilde{P}, b\widetilde{P}, c\widetilde{P})$, $\mathscr{B}$ decides whether $c\widetilde{P} = ab\widetilde{P}$ holds or not.

- Setup: $\mathscr{B}$ firstly sets $P_1 = P, \widetilde{P}_1 = \widetilde{P}, \widetilde{P}_{tm} = b\widetilde{P}, P_{link} = bP$. Then, other related parameters of CA and RSU are produced as the actual execution.
- Public key oracle: When $\mathscr{A}$ requests the public key of OBUs, $\mathscr{B}$ returns $P_{obu} = x_{obu}P_1$ to $\mathscr{A}$, where x_{obu} is selected randomly from Z_q^*.
- Challenge: $\mathscr{A}$ chooses two public key P_{obu_0}, P_{obu_1} obtained from the public key oracle and a message m to challenge, where $P_{obu\_0} \neq P_{obu\_1}$. $\mathscr{B}$ tosses a coin $b \in 0, 1$ and $\mathscr{B}$ responds as follows: $\mathscr{B}$ firstly selects $r \in Z_q^*$ randomly, then, it outputs the signature $\{\tau_1, \tau_2, \widetilde{\tau}_3, \widetilde{\tau}_4, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9\}$ as follows:

$$\tau_1 = x_{rsu}P_2 - r\cdot(x_{ca}P_{obu\_b}), \quad \tau_2 = rP_1, \quad \widetilde{\tau}_3 = a\widetilde{P} = a\widetilde{P}_1,$$

$$\widetilde{\tau}_4 = x_{obu\_b}\widetilde{P}_1 + c\widetilde{P}, \quad \tau_5 = x_{obu\_b}\sigma_2, \quad \tau_6 = r \cdot aP = a\tau_2,$$
$$\tau_7 = x_{obu\_b}H_1(m), \quad S_1 = \tau_9\tau_2 + \tau_8\tau_5,$$
$$S_2 = \tau_9 H_1(m) + \tau_8 \cdot \tau_7$$

where x_{obu_b} is the private key corresponding to P_{obu_b}.

- Signature oracle: Upon receiving the query on m and $P_{obu} \notin P_{obu\_0}, P_{obu\_1}$ from $\mathscr{A}$ obtained from the public key oracle, $\mathscr{B}$ first selects $r, \alpha \in Z_q^*$ randomly, then it computes and outputs the signature $\{\tau_1, \tau_2, \widetilde{\tau}_3, \widetilde{\tau}_4, \tau_5, \tau_6, \tau_7, \tau_8, \tau_9\}$ as follows.

$$\tau_1 = x_{rsu}P_2 - r \cdot (x_{ca}P_{obu}), \quad \tau_2 = rP_1, \quad \widetilde{\tau}_3 = \alpha\widetilde{P}_1,$$
$$\widetilde{\tau}_4 = x_{obu}\widetilde{P}_1 + \alpha\widetilde{P}_{tm}, \quad \tau_5 = rP_{obu}, \quad \tau_6 = \alpha\tau_2,$$
$$\tau_7 = x_{obu}H_1(m), \quad S_1 = \tau_9\tau_2 + \tau_8\tau_5,$$
$$S_2 = \tau_9 H_1(m) + \tau_8 \cdot \tau_7$$

where x_{obu} is the private key with respect to P_{obu}.

- Output: $\mathscr{A}$ outputs its guess $b' \in \{0, 1\}$. If $b' = b$, $\mathscr{B}$ outputs 1 which means $c\widetilde{P} = ab\widetilde{P}$, otherwise, outputs 0.

Note because $\mathscr{B}$ does not abort at any step in all simulations, we can know that the overall probability of success for $\mathscr{B}$ is the same as the probability of success for $\mathscr{A}$. Therefore, our proposed signature scheme is anonymous because the eDDH problem is hard problem.

- Threshold authentication: To ensure the authenticity of some special messages, such as traffic accident message, threshold based trust mechanism is adopted in our proposed scheme. i.e., a single signed message would not be accepted by the receiver unless the number of received signatures from different senders on the same message has exceeded the threshold number.
  - —since the values $\tau_1 = c_1 - r'(x_{obu}P_{ca})$ and $\tau_2 = c_2 + r'P_1$ of τ are derived from the group certificate $cert_g = (c_1, c_2)$, τ_1 and τ_2 can be used to check whether the sender poses a group certificate of a RSU, and the values $\widetilde{\tau}_3 = \alpha \cdot \widetilde{P}_1$, $\widetilde{\tau}_4 = x_{obu} \cdot \widetilde{P}_1 + \alpha \cdot \widetilde{P}_{tm}$ which are computed by using the private key of the sender x_{obu} can be employed to trace the identity of signer.
  - —However, the value $\tau_7 = x_{obu}H_1(m)$ in τ is generated by the private key x_{obu} and the hash value of m, thus the receiver can use the value τ_7 to determine whether the received signatures for the message m are produced by the same signer. If a malicious sender attempts to break the threshold mechanism by producing multiple different signatures for the same message m, this misbehavior would be detected by the receiver. Therefore, threshold authentication is achieved.
- Controllable linkabilty: The controllable linkability in our scheme refers to that any other entities except the service provider SP cannot link any two or messages to a sender, i.e., only SP can determine whether two anonymous signatures are produced by the same sender. Since the signature is produced by using random numbers, and the group certificate of a signer is randomized before it is assigned each time. Even if two or more signatures about different messages are generated by the same user, an adversary $\mathscr{A}$ cannot figure out whether they are signed by the same user, which means that the exchanged messages are unlinkable for the outside adversary. However, SP with a linking key can determine whether two different signatures are produced by the same sender, thus it can

provide personalized services. Therefore, the controllable linkability has been achieved in our presented anonymous authentication scheme.

- Non-repudiation: Although our presented anonymous authentication scheme enable the signer of a message to be anonymous, the sender cannot deny the signature, and the non-repudiation of the proposed scheme is still effective. Every broadcasted signature message consists of a dynamic pseudonyms and a dynamic group certificate, which is computed by using the public key and private key of the signer, group certificate and random numbers. According to Theorem 2, an adversary $\mathscr{A}$ cannot produce a valid signature if he or she does not have a private key or the corresponding group certificate. Therefore, the user can never deny the broadcasted signature message generated by its private key and group certificate. Thus, the non-repudiation of our scheme is achieved.
- Conditional traceability: The conditional traceability in our presented scheme means the message sender cannot be traced by any unauthorized entity, and the identity of a signature sender can only be derived by the trusted third party (TTP). Since the public key of the signer is used to produce the values $\widetilde{\tau}_3, \widetilde{\tau}_4$ in signature by utilizing CA's public key, CA is the only entity who can recover the identity corresponding to the signature. Suppose there exists an adversary $\mathscr{A}$ can violate the conditional traceability of our presented authentication scheme, thus $\mathscr{A}$ can produce a valid signature without the private key or the corresponding group certificate, which is contradicts with Theorem 2. Therefore, our proposed scheme achieves traceability.

6 Performance Evaluation

In this section, we give out a comparison on the computation cost and communication overhead with existing group signature based authentication schemes. In addition, the running time of compared schemes are evaluated with implementation based on cryptographic libraries.

6.1 Computation Cost

We focus on the computation cost of the signing process and the verifying process with the existing similar schemes [15,17,28,29], and the cost of revocation check is not considered since this function is our specific goal and the length of the revocation list is uncertain. The related computation cost is summarized in Tab. 3. Since the time of transmission depends on the real network, not the concret scheme, it is not considered in the comparison.

For convenience, some notations are defined as follows:

- T_{bp}: The running time of one bilinear pairing operation.
- T_{mul}: The running time of one ECC-based scalar point multiplication operation.
- T_{ex}: The running time of one exponentiation operation.

In sign stage, when a signer signs a single message, the computation cost in scheme [15,17] is about $11T_{ex}$; the computation cost in scheme [28] is about $10T_{ex}$; the computation cost in scheme [29] is about $9T_{ex}$; the computation cost in our proposed scheme is about $8T_{mul}$. In addition, in the verification stage, when a verifier verifies a single message, the computation cost in scheme [15,17] is about $11T_{ex} + 1T_{bp}$; the computation cost in scheme [28] is about $4T_{ex} + 10T_{bp}$; the computation cost in scheme [29] is about $4T_{ex} + 9T_{bp}$; the computation cost in our proposed scheme is about $4T_{mul} + 6T_{bp}$.

However, a verifier may need to verify multiple messages in a verification period, thus we assume that the traffic density is n which is the verifier received the number of messages in a verification period. Therefore, when a verifier verifies n messages simultaneously, the computation

cost in scheme [15,17] is about $11nT_{ex} + nT_{bp}$; the computation cost in scheme [28] is about $4nT_{ex} + (n+5)T_{bp}$ in batch; the computation cost in scheme [29] is about $4nT_{ex} + (n+6)T_{bp}$ in batch; the computation cost in our proposed scheme is about $4nT_{mul} + (n+4)T_{bp}$ in batch.

All compared schemes are implemented based on the JPBC library and OpenSSL library, the execution time of all basic operations listed in Tab. 2, and the experiments are constructed on a Windows 10 PC with an Intel(R) Core(TM) i7-6500U CPU.

Table 2: Execution times of the basic operation

Operation	T_{ex}	T_{mul}	T_{bp}
Execution	0.483	0.326	6.280

Table 3: Computation comparison

Scheme	Sign cost	Verify cost of n messages
Reference [15]	$11T_{ex}$	$11nT_{ex} + nT_{bp}$
Reference [17]	$11T_{ex}$	$11nT_{ex} + nT_{bp}$
Reference [28]	$10T_{ex}$	$4nT_{ex} + (n+6)T_{bp}$
Reference [29]	$9T_{ex}$	$4nT_{ex} + (n+5)T_{bp}$
Proposed	$8T_{mul}$	$4nT_{mul} + (n+4)T_{bp}$

As shown in Figs. 3 and 4, we present an intuitive comparison on the execution time of each scheme in sign stage and in verify stage respectively. For example, to verify 100 signatures, the required time in scheme [15,17] is about 1159 ms ($= 11nT_{ex} + nT_{bp} = 11 * 0.483 * 100 + 100 * 6.28$); the value in [28] is about 852 ms ($= 4nT_{ex} + (n+5)T_{bp} = 4 * 0.483 * 100 + (100+5) * 6.28$); the value in [29] is about 858 ms ($= 4nT_{ex} + (n+6)T_{bp} = 4 * 0.483 * 100 + (100+6) * 6.28$); which implies that the value in our proposed scheme is about 783 ms ($= 4nT_{mul} + (n+4)T_{bp} = 4 * 0.326 * 100 + (100+5) * 6.28$). Therefore, we can know our proposed scheme is more effective than other existing schemes for vehicle sensor networks according to the above analysis.

6.2 Communication Overhead

In this subsection, we evaluate the communication overhead with the existing group signature based schemes [15,17,28,29]. The related comparison result is summarized in Tab. 4. In Tab. 4, ℓ_G, ℓ_q and ℓ_H represent the bit-length of an element of group G, the order of group G and an element of hash H, respectively.

In the experiment, we choose SECG-160 curve and hash function SHA512 to simulate these operations. Due to the security level of SECG-160 curve is almost equivalent to the RSA 1024 bit, we set $\ell_G = 1024$ when we simulate exponentiation operation in G. When $\ell_G = 320$, $\ell_q = 160$ and $\ell_H = 512$, the signature length of our proposed scheme is almost $2912 (= 7 * 320 + 512 + 160)$ bits or 364 bytes. To have equivalent security level, we let $\ell_G = 1024$, $\ell_q = 1023$ for these schemes [15,17,28,29] since they use exponentiation operation in G. Therefore, we can compute the signature length of scheme [17] is almost 8699 bits ($= 3 * 1024 + 512 + 5 * 1023$) or 1088 bytes; the signature length of scheme [17] is almost 7676 bits($= 3 * 1024 + 512 + 4 * 1023$) or 960 bytes; the signature length of scheme [28] is almost 8703 bits ($= 7 * 1024 + 512 + 1 * 1023$) or 1088

bytes; the signature length of scheme [29] is almost 10751 bits $(= 9 * 1024 + 512 + 1 * 1023)$ or 1343 bytes.

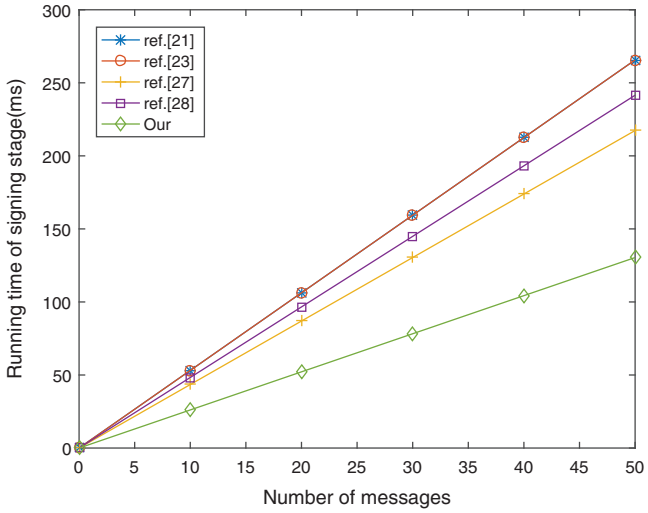

Figure 3: Computation cost of signing stage

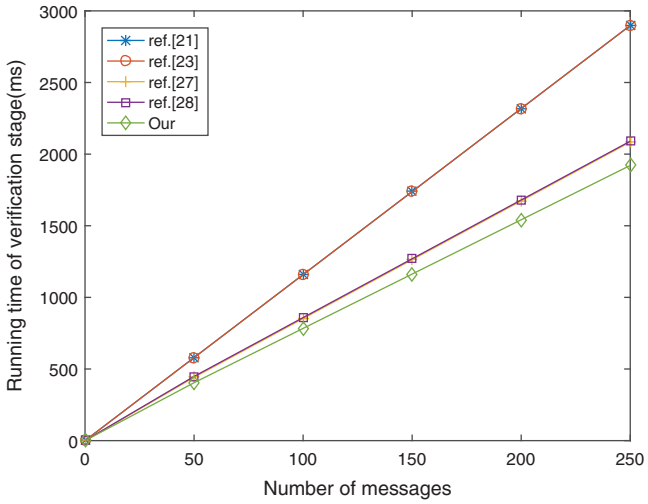

Figure 4: Computation cost of verification stage

An intuitive comparison on communication overhead in term of the number of messages is given in Fig. 5. It can be seen that the communication overhead increases linearly with the growth of the number of messages transmitted. Based on the comparison above, we can conclude that the communication overhead of our presented anonymous authentication scheme is relatively low.

Table 4: The comparison of signature length

Scheme	Signature length
Reference [15]	$3\ell_G + 1\ell_H + 5\ell_q$
Reference [17]	$3\ell_G + 1\ell_H + 4\ell_q$
Reference [28]	$7\ell_G + 1\ell_H + 1\ell_q$
Reference [29]	$9\ell_G + 1\ell_H + 1\ell_q$
Proposed	$7\ell_G + 1\ell_H + 1\ell_q$

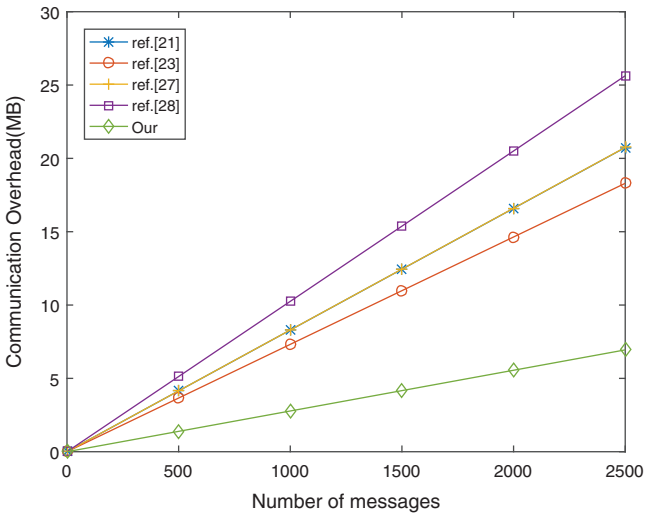

Figure 5: Comparison of communication overhead

7 Conclusion

In this paper, a group signature-based anonymous authentication scheme with controllable linkability was proposed. The scheme is designed to enable providers who have a linking key to determine whether two messages were produced by the same signer, while preserving the user's anonymity. Threshold authentication enables the receiver to figure out whether the received signature is produced by the same sender to prevent the replay attack. In addition, the function of verifier-local revocation is supported (i.e., a verifier is able to check whether a received signature is generated by a revoked user). Security and performance evaluations demonstrated the utility of our presented scheme.

Funding Statement: Our work was jointly supported by the National Natural Science Foundation of China (Nos. 61872051, 61702067), the Venture & Innovation Support Program for Chongqing Overseas Returnees (No. CX2018122), the Chongqing Natural Science Foundation (No. cstc2020jcyjmsxmX0343).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

1. Hubaux, J. P., Capkun, S., Luo, J. (2004). The security and privacy of smart vehicles. *IEEE Security & Privacy, 3(2),* 49–55. DOI 10.1109/MSP.2004.26.
2. Chuang, M. C., Lee, J. F. (2014). TEAM: Trust-extended authentication mechanism for vehicular ad hoc networks. *IEEE International Conference on Consumer Electronics, 8(3),* 749–758.
3. Okhovvat, M., Kangavari, M. R. (2019). A mathematical task dispatching model in wireless sensor actor networks. *Computer Systems Science and Engineering, 34(1),* 5–12.
4. Zhou, Y. S., Zhao, X. F., Jiang, Y., Shang, F. J., Deng, S. et al. (2017). An enhanced privacy-preserving authentication scheme for vehicle sensor networks. *Sensors, 17(12),* 28–54.
5. Abdelatif, S., Derdour, M., Ghoualmi, N. (2020). VANET: A novel service for predicting and disseminating vehicle traffic information. *Sensors, 33(6),* 2366.
6. Huang, X., Xiang, Y., Chonka, A., Deng, R. H. (2011). A generic framework for three-factor authentication: Preserving security and privacy in distributed systems. *IEEE Transactions on Parallel & Distributed Systems, 22(8),* 1390–1397. DOI 10.1109/TPDS.2010.206.
7. Li, J., Lu, H., Guizani, M. (2015). ACPN: A novel authentication framework with conditional privacy–preservation and non-repudiation for VANETs. *IEEE Transactions on Parallel & Distributed Systems, 26(4),* 938–948. DOI 10.1109/TPDS.2014.2308215.
8. Hao, H., Lu, R., Cheng, H. (2016). TripSense: A trust-based vehicular platoon crowdsensing scheme with privacy preservation in VANETs. *Sensors, 16(6),* 803. DOI 10.3390/s16040562.
9. Wang, H., Qin, B., Wu, Q., Domingo-Ferrer, J. (2017). TPP: Traceable privacy-preserving communication and precise reward for vehicle-to-grid networks in smart grids. *IEEE Transactions on Information Forensics & Security, 10(11),* 2340–2351. DOI 10.1109/TIFS.2015.2455513.
10. Bohli, J. M., Pashalidis, A. (2011). Relations among privacy notions. *ACM Transactions on Information & System Security, 14(1),* 362–380.
11. Zhao, D., Peng, H., Li, L., Yang, Y. (2014). A secure and effective anonymous authentication scheme for roaming service in global mobility networks. *Wireless Personal Communications, 78(1),* 247–269. DOI 10.1007/s11277-014-1750-y.
12. Okhovvat, M., Kangavari, M. R. (2019). Tslbs: A time-sensitive and load balanced scheduling approach to wireless sensor actor networks. *Computer Systems Science and Engineering, 34(1),* 13–21.
13. Arafatur, M., Rahman, A. (2020). A scalable hybrid MAC strategy for traffic-differentiated IoT-enabled intra-vehicular networks. *Computer Communications, 157(8),* 320–328. DOI 10.1016/j.comcom.2020.04.035.
14. Fayyad, U. M., PiatetskyShapiro, G., Smyth, P. (1996). From data mining to knowledge discovery: An overview. *Advances in Knowledge Discovery and Data Mining, 17(3),* 1–34.
15. Hwang, J. Y., Lee, S., Chung, B. H., Cho, H. S., Nyang, D. H. (2011). Short group signatures with controllable linkability. *2011 Work on Lightweight Security & Privacy: Devives, Protocols, and Applications,* Istanbul, 44–52.
16. Hwang, J. Y., Lee, S., Chuang, B. H., Cho, H. S., Nyang, D. H. (2013). Group signatures with controllable linkability for dynamic membership. *Information Sciences, 222(3),* 761–778. DOI 10.1016/j.ins.2012.07.065.
17. Hwang, J. Y., Chen, L., Cho, H. S., Nyang, D. H. (2015). Short dynamic group signature scheme supporting controllable linkability. *IEEE Transactions on Information Forensics & Security, 10(6),* 1109–1124. DOI 10.1109/TIFS.2015.2390497.
18. Liu, A. F., Min, J., Ota, K., Zhao, M. (2018). Reliable differentiated services optimization for network coding cooperative communication system. *Computer Systems Science and Engineering, 33(4),* 235–250.
19. Raya, M., Hubaux, J. P. (2007). Securing vehicular ad hoc networks. *Journal of Computer Security, 15(1),* 39–68. DOI 10.3233/JCS-2007-15103.
20. Lu, R., Lin, X., Zhu, H., Ho, P. H., Shen, X. (2008). ECPP: Efficient conditional privacy preservation protocol for secure vehicular communications. *27th IEEE International Conference on Computer*

Communications, Joint Conference of the IEEE Computer and Communications Societies, Phoenix, AZ, USA, 13–18 April 2008, 1229–1237.

21. Huang, X., Mu, Y., Susilo, W., Wong, D. S., Wu, W. (2012). Certificateless signatures. *Computer Journal, 55(4),* 457–474. DOI 10.1093/comjnl/bxr097.
22. Dan, B., Boyen, X., Shacham, H. (2004). Short group signatures. *Advances in Cryptology, 15(19),* 41–55.
23. Chaurasia, B. K., Verma, S., Bhasker, S. M. (2008). Message broadcast in VANETs using group signature. *2008 Fourth International Conference on Wireless Communication and Sensor Networks,* Allahabad, 131–136.
24. Lin, X., Sun, X., Ho, P. H., Shen, X. (2007). GSIS: A secure and privacy-preserving protocol for vehicular communications. *IEEE Transactions on Vehicular Technology, 56(6),* 3442–3456. DOI 10.1109/TVT.2007.906878.
25. Harn, L. (2013). Group authentication. *IEEE Transactions on Computer, 62(9),* 1893–1898. DOI 10.1109/TC.2012.251.
26. Zhang, L., Wu, Q., Solanas, A., Domingo-Ferrer, J. (2010). A scalable robust authentication protocol for secure vehicular communications. *IEEE Transactions on Vehicular Technology, 59(4),* 1606–1617. DOI 10.1109/TVT.2009.2038222.
27. Morshed, M. M., Atkins, A., Yu, H. (2012). Efficient mutual authentication protocol for radiofrequency identification systems. *IET Communications, 6(16),* 2715–2724. DOI 10.1049/iet-com.2011.0807.
28. Shao, J., Lu, R., Lin, X. X., Zou, C. (2015). New threshold anonymous authentication for VANETs. *IEEE/CIC International Conference on Communications in China,* Shenzhen, China, 1–6.
29. Shao, J., Lin, X., Lu, R., Zou, C. (2016). A threshold anonymous authentication protocol for VANETs. *IEEE Transactions on Vehicular Technology, 65(3),* 1711–1720. DOI 10.1109/TVT.2015.2405853.