PEACS: A Privacy-Enhancing and Accountable Car Sharing System

Yulin Liu, Debiao He, *Member, IEEE*, Zijian Bao, Min Luo, and Cong Peng

***Abstract*—Car sharing is gaining increased popularity in urban transportation which allows individuals to conveniently rent vehicles for short periods. Such systems, however, present considerable challenges to security, such as unauthorized access and privacy data breaches, as service providers are able to track the precise mobility patterns of all customers. Nevertheless, efforts in solving the security and privacy concerns associated with car-sharing services are relatively few, in particular for a tradeoff between privacy preservation and accountability of misbehaviors. In this work, we propose an efficient privacy-enhancing and accountable car sharing system (PEACS), introduced to mitigate the aforementioned threats. PEACS primarily employs several key ingredients, including structure-preserving signatures on equivalence classes (J CRYPTOL's 19), as well as two primitives designed in this article, namely, signatures of knowledge and identity-based structure-preserving signatures with a tag on equivalence classes. Furthermore, we employ a bivariate polynomial function to establish a revocation mechanism that ensures accountability. We provide thorough security proof to demonstrate the security and privacy of PEACS. Comprehensive performance evaluation and comparison results point out that our proposed scheme is feasible in practical settings.**

***Index Terms*—Accountablility, car sharing, privacy preserving, signatures of knowledge (SoK), structure-preserving signatures.**

Manuscript received 25 October 2023; revised 19 December 2023; accepted 7 January 2024. Date of publication 12 January 2024; date of current version 25 April 2024. This work was supported in part by the Shandong Provincial Key Research and Development Program under Grant 2021CXGC010107; in part by the National Natural Science Foundation of China under Grant U21A20466, Grant 62172307, and Grant 62272350; in part by the New 20 Project of Higher Education of Jinan under Grant 202228017; and in part by the Fundamental Research Funds for the Central Universities under Grant 2042023KF0203. *(Corresponding authors: Debiao He; Cong Peng.)*

Yulin Liu is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Institute of Information Technology, Shenzhen Institute of Information Technology, Shenzhen 518172, China (e-mail: liuyulin@whu.edu.cn).

Debiao He is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250014, China (e-mail: hedebiao@163.com).

Zijian Bao and Cong Peng are with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China (e-mail: baozijian@whu.edu.cn; cpeng@whu.edu.cn).

Min Luo is with the School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, and also with the Shanghai Technology Innovation Centre of Distributed Privacy-Preserving Artificial Intelligence, Matrix Elements Technologies, Shanghai 200232, China (e-mail: mluo@whu.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3353458

I. Introduction

CAR SHARING, one of the main types of shared mobility services, is growing in popularity as a convenient, cost-effective alternative to traditional car ownership since it enables users to share their vehicles in a simple manner. The cars can be company- or individual-owned. Generally speaking, car sharing systems fall into one of two sharing models: 1) station-based (e.g., Cambio); and 2) free-floating car sharing (e.g., Enjoy[1]). When it comes to station-based car sharing, cars are picked up at a car-sharing parking station and returned to either the same place or a different parking station from their origin. While cars are rented and returned virtually anywhere in the free-floating car-sharing model [1].

Recent report [2] has shown that the number of users is predicted to reach 62.1 million by 2027 worldwide in the car-sharing segment, which means car sharing business is booming dramatically. The most critical reason is that vehicle sharing leads to a decrease in the number of automobiles, which brings substantial benefits to society and climate, such as reducing city congestion [3], the demand for parking space [4], and urban emissions [5]. As a new form of business to effectively realize sustainable consumption, car sharing is of interest to both academic research and practical applications.

Despite these advantages, car sharing systems collect a vast amount of sensitive information, posing risks not only to system security but also to user privacy. For example, Enev [6] demonstrated that by analyzing a 15-min driving pattern, 15 drivers could be distinguished with an accuracy ranging from 87% to 99%. Before a user enjoys car-sharing services, the user needs to provide his identity information and prove that he has the driving ability for service providers (SPs), e.g., photography of his driving license. As a result, SPs, or other entities who gain access to detailed driving records can inevitably infer sensitive information about renters' activities, and link two car-sharing requests made by the same renter to deduce renters' preferences (through rental time, pick-up/return location, duration of use) [7], [8], [9]. This poses a detriment to user privacy and may somewhat entail limited applicability of car sharing. The new EU general data protection regulation (GDPR) [10] clearly defines protections for the collection and processing of sensitive personal data.

While safeguarding user privacy, accountability for misbehaviors should still be maintained under specific

[1] https://enjoy.eni.com/

circumstances, e.g., some renters refuse to return shared vehicles (SVs) on time or damage vehicles intentionally or unintentionally while using them. This introduces challenges in addressing a tradeoff between privacy and accountability in car sharing systems. Thus, it is vital to design a mechanism for car sharing services that provide robust privacy and accountability guarantees.

A. Related Work

Several works have been performed on car sharing systems, but there exist few studies in the area of privacy and security for car sharing systems [8], [9], [11], [12], [13], [14], [15]. The SECREDAS EU project [13] develops a reference architecture integrating security and privacy for car-sharing systems. Symeonidis et al. first proposed a keyless car sharing system with privacy considerations. Taking into account the designed model and functional prerequisites, they conducted a comprehensive threat analysis of car sharing systems, including threats to user privacy [11]. Afterward, they presented SePCAR to provide security and privacy guarantees in car sharing system [12], which designs a decentralized protocol for car access tokens distribution and adopts time-consuming multiparty computation (MPC) protocols for the accountability protecting against misbehaviors. Moreover, they extended SePCAR to HERMES to achieve a scalable and more efficient car sharing system [9]. Akash et al. used smart contract technology to deal with the booking and payment aspects of car sharing services [14]. By utilizing a two-factor authentication protocol and an RFID card, Dmitrienko and Plappert [15] designed a novel free-floating car sharing system. However, they all concentrated on the free-floating car sharing model. Since network coverage and data service quality vary greatly in different locations, the station-based mode is more widely used than the free-floating mode for reliability considerations.

To achieve both privacy preservation and accountability in vehicular services systems, current accountable anonymous mechanisms come in two flavors: 1) pseudonym-based [16], [17], [18] and 2) group-signature-based [8], [19], [20]. Pseudonym-based protocols require frequent pseudonym preloading or renewal from online authorities, resulting in significant bandwidth and storage overhead. Group signature-based approaches appear to be a more promising option, compared to the pseudonym-based methods, for efficient authentication in vehicular services systems. Thus, our emphasis lies on group signature-based approaches, which enable users to protect their privacy by concealing true identities within a group through indistinguishable signatures, while also allowing authorities to revoke the membership of malicious users. However, current group signature-based schemes [8], [19], [20] fail to strike a proper balance between anonymity, accountability, and efficiency, as they either impose substantial computational overhead during verification or lead to significant communication costs.

B. Contributions

Our work is closely relevant to DAPA [8] introduced by Huang et al. They designed a car sharing system under a decentralized identity management scheme. The validation servers managed the identities of customers in a dynamic and distributed way. There are private channels between the client and all verification servers (VSs), as well as between VSs. Since the VSs are dynamically changing, the channels need to be established and dismantled frequently. Besides, it further causes high-communication overheads. This may be not desirable for practical use. Thereby, the following open question is raised:

"*Is it possible to construct a secure station-based car sharing system providing anonymity, unlinkability, and accountability notions efficiently?*"

We answer the question posed above to the affirmative by proposing an efficient privacy-enhancing and accountable car sharing system (PEACS). Different from previous car-sharing scenarios, we introduce a Two-Tier anonymous credential model that requires multiparty [i.e., parking lots (PLs)] approval and supervision, thereby increasing trust in the verification of the authenticity and integrity of renters' requests. In summary, the main contributions of our work are listed in points as follows.

1) Two Efficient, Well-Designed Cryptographic Primitives: PEACS includes an identity-based structure-preserving signature scheme with a tag on equivalence classes (ITEQ) and signatures of knowledge (SoK). In brief, one can publicly update a message and a ITEQ signature, with the newly generated signature being indistinguishable from the original. The SoK supports conditional anonymous authentication, letting renters access car-sharing without privacy concerns. It also prevents credential lending and allows trusted authority (TA) to trace malicious renters.

2) Design and Prototype of PEACS: PEACS is a station-based car-sharing system that prioritizes privacy while ensuring accountability. PEACS employs structure-preserving signatures on equivalence classes ($\mathsf{SPS\text{-}EQ}$) scheme [21], two aforementioned building blocks (namely, SoK and ITEQ), as well as a revocation mechanism via a bivariate polynomial. Within the car-sharing system, $\mathsf{SPS\text{-}EQ}$ and ITEQ signatures are used to hide renters' public keys and credentials. In the two-tier anonymous credential model, renters first acquire anonymous credentials (i.e., $\mathsf{SPS\text{-}EQ}$) from the SP. After a legitimate inspection, they receive supplementary credentials (i.e., ITEQ) from PLs. We opt for $\mathsf{SPS\text{-}EQ}$ to improve efficiency, as it allows for showing a credential without requiring zero-knowledge proofs for proving the well-formedness, thanks to its randomizability. Meanwhile, the aggregability of ITEQ precisely fulfills the requirements for multiparty supervision.

3) Comprehensive Performance Evaluation: We perform extensive simulation to demonstrate the feasibility of PEACS. In contrast to existing approaches, experimental results show that PEACS achieves a better balance between important properties, such as anonymity, unlinkability, accountability,

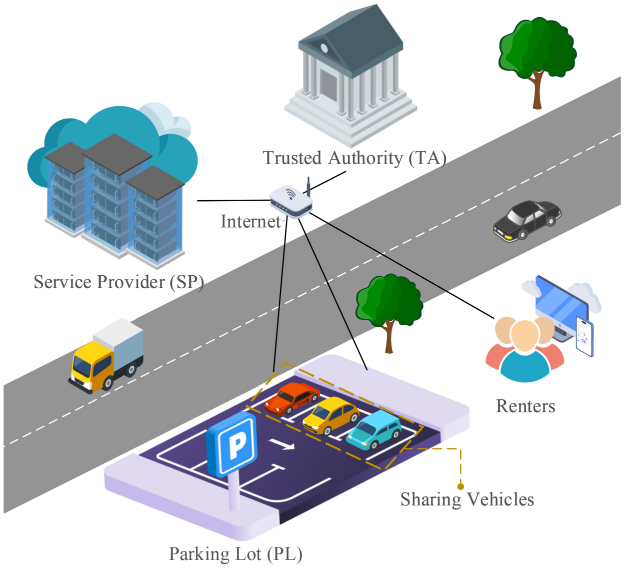

Fig. 1. System architecture.

and efficiency in terms of computation and communication costs.

II. Problem Formulation

A. System Model

As it is depicted in Fig. 1, PEACS includes five entities: 1) TA; 2) SPs; 3) SVs; 4) renters; and 5) PLs.

1) *TA:* This entity is the trusted entity that is responsible for the system initialization, the registration service of SPs and PLs, and the accountability.
2) *SPs:* SPs are platforms that provide car sharing services. They account for verifying the validity of renters' requests, including whether the credentials possessed are issued by themselves, etc. If all pass, SPs assign appropriate vehicles for valid renters and settle the fee payable by the renters based on time if the SVs are properly returned.
3) *SVs:* SVs are dispersedly dropped into different PLs by SPs. Renters only with correct codes can drive the assigned SVs.
4) *Renters:* Renters are customers of car sharing platforms who request access to SVs. To be eligible for vehicle rental, renters are required to supply SPs with valid drivers' license, insurance, and other details for identity verification, and renters will obtain credentials issued from SPs to get rental services if the verification passes.
5) *PLs:* PLs are the locations where renters pick up and return SVs. They need to register with TA to obtain anonymous identities with the corresponding secret keys to sign renters' credentials.

B. Threat Model

TA serves as a fully trusted entity thus it is infeasible for TA to be corrupted. SVs are trusted, as they possess hardware security modules (HSMs) with tamper-resistant storage. The adversaries are classified into two types: 1) internal adversaries and 2) external adversaries, the difference between the two is whether they have been registered and authenticated in car sharing systems.

For internal adversaries, renters are active adversaries, and SPs are passive adversaries (i.e., honest-but-curious). For instance, renters may illegally access vehicles, tamper rental records of profits, and destroy SVs. Also, a malicious renter may perform relay attacks: attempting to use a signature previously obtained from a PLs to deceive SP's verification after failing to return a vehicle on time. SPs are semi-honest, they execute the procedure honestly but they seek to learn some sensitive data, i.e., identities. All internal adversaries may try to get the real identities of renters. External adversaries eavesdrop on communication channels, trying to expose the origin, destination, and private preferences of renters from accessed locations.

C. Security and Privacy Goals

This part highlights the security and privacy requirements that are listed below to prevent malicious behaviors.

Security:

1) *Entity Authentication:* Each authorized participant within the system should demonstrate that they are the actual entities conducting the actions.
2) *Accountability:* It is of vital importance to provide accountability (i.e., traceability and revokability) to deter misbehavior during vehicular services. TA can identify misbehaved renters and revoke them when needed. Renters are unable to get rental services with the revoked privilege.
3) *Nonframeability:* No renters can frame others by fooling the tracing procedure.

Privacy:

1) *Anonymity:* The identities of renters can be concealed from both internal and external adversaries if they honestly follow the vehicular rental procedure.
2) *Unlinkability:* Given two car rental service requests made by the same renter using the same credential, adversaries cannot link these renting records of any renter. This property protects renters' access records from being correlated. Furthermore, it is necessary to ensure that in the same rental transaction, adversaries cannot link a user with his corresponding trajectory to guarantee trip privacy.
3) *Location Privacy:* Location privacy specifically refers to the privacy of the SV's pick-up and drop-off parking locations. The travel trajectories of renters must be protected from other entities. Namely, no adversaries can reveal the corresponding relationship among a renter, the PL to pick up PL_p, and the PL to return PL_r.

III. Background

We illustrate the preliminaries involved and recap the required cryptographic primitives in this section. To clarify, we present the frequently used notations in Table I.

Definition 1 (Bilinear Map): Let $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ be finite groups of the same prime order q. $g, \widetilde{g}, g_T$ are the generators

TABLE I
FREQUENTLY USED NOTATIONS

Notations	Descriptions
κ	A security parameter
q	A large prime
$g, \widetilde{g}, g_T$	Generators of $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, respectively
$\hat{e}$	A bilinear map
$\mathbb{BG}$	A Type-3 bilinear group
$\mathcal{H}, \mathcal{H}_{1,\ldots,4}$	Secure cryptographic hash functions
$\mathsf{MPK}, \mathsf{msk}$	The master public/secret key of the system
$\vec{\mathsf{pk}}_{\mathcal{Q}}, \vec{\mathsf{sk}}_{\mathcal{Q}}$	The public/secret key of a service provider
uid	The identity of a renter
$\mathsf{upk}, \mathsf{usk}$	The public/secret key of a renter
nym	Anonymous identity of a renter
$\mathsf{aid}, \mathsf{rid}$	Anonymous/real identity of a parking lot
$\mathsf{sk}_{\mathsf{ID}}$	The secret key of a parking lot
$cred$	A credential of a renter
τ	A time tag
Ψ	A signature of knowledge
$\mathcal{F}(\mathcal{X}, \mathcal{Y})$	A bivariate polynomial
θ_{ij}	The coefficient of a bivariate polynomial $\mathcal{F}(\mathcal{X}, \mathcal{Y})$
$\Theta_i(\mathcal{Y})$	A polynomial fragment of renter u_i, which is equivalent to $\mathcal{F}(u_i, \mathcal{Y})$
$\Delta(\mathcal{X}), \Gamma(\mathcal{X})$	The revocation/broadcast polynomial
$(a_i)_{i\in[\ell]}$	The array of $\{a_1, \ldots, a_\ell\}$
$\mathcal{PPT}$	The abbreviation of probabilistic polynomial time

of $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$, respectively. $\hat{e}$ is an efficiently computable bilinear map $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with the following properties.

1) *Bilinearity:* For $\phi, \psi \in \mathbb{Z}_q^*$ and all $u \in \mathbb{G}_1, \widetilde{v} \in \mathbb{G}_2$, $\hat{e}(u^{\phi}, \widetilde{v}^{\psi}) = \hat{e}(u, \widetilde{v})^{\phi\psi}$.
2) *Nondegeneracy:* There exist some element $u \in \mathbb{G}_1, \widetilde{v} \in \mathbb{G}_2$ satisfies $\hat{e}(u, \widetilde{v}) \neq g_T$.

We only consider Type-3 pairings [22] in our work. To ease the readability we marked all the elements on $\mathbb{G}_2$ with "tilde."

Definition 2 (Discrete Logarithm (DL) Assumption): With a bilinear group $\mathbb{BG} = (\hat{e}, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, \widetilde{g}, g_T)$, the DL Assumption in $\mathbb{G}_i$ states for all nonuniform $\mathcal{PPT}$ adversary $\mathcal{A}$, we have that the probability $\Pr[g_i \leftarrow \mathbb{G}_i, y \leftarrow \mathbb{Z}_q, y' \leftarrow \mathcal{A}(\mathbb{BG}, g_i^y) : y' = y]$ is negligible.

Definition 3 (Decisional Diffie–Hellman (DDH) Assumption): With a bilinear group $\mathbb{BG} = (\hat{e}, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, \widetilde{g}, g_T)$, the DDH Assumption in $\mathbb{G}_i$ states for all nonuniform $\mathcal{PPT}$ adversary $\mathcal{A}$, we have that the probability $(\Pr[b \leftarrow \{0,1\}, r, s, t \leftarrow \mathbb{Z}_q, b' \leftarrow \mathcal{A}(\mathbb{BG}, g_i^r, g_i^s, g_i^{(1-b)\cdot t + b\cdot rs}) : b' = b] - (1/2))$ is negligible.

1) Noninteractive Zero-Knowledge Argument Systems: Noninteractive zero-knowledge (NIZK) argument [23] allows a user to convince others of the correctness of a statement without revealing anything else noninteractively. For an NP-language $\mathcal{L}_{\mathcal{R}}$ with relation $\mathcal{R} = (x, w)$, where x is a public statement and w is a private witness. We require NIZK systems, constructed via Fiat–Shamir transform [24] on a Σ-protocol [25], to be complete, sound, and zero-knowledge.

2) Signatures of Knowledge: SoK, as a type of NIZK arguments [26], also serve as digital signatures. An SoK scheme for $\mathcal{L}_{\mathcal{R}}$ is specified as follows.

Definition 4 (SoK): An SoK scheme is made up of the following algorithms $\Pi_{\mathcal{S}} = (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Vrf})$.

1) $\mathsf{params}_{\mathcal{S}} \leftarrow \mathsf{Setup}(1^{\kappa})$: Takes a security parameter 1^{κ} as input and outputs public parameters $\mathsf{params}_{\mathcal{S}}$.
2) $\Psi \leftarrow \mathsf{Sign}(\mathsf{params}_{\mathcal{S}}, x, w, M)$: Takes public parameters $\mathsf{params}_{\mathcal{S}}$, a public statement x, a private witness w, and a message M as input, and outputs a signature Ψ.
3) $b \leftarrow \mathsf{Vrf}(\mathsf{params}_{\mathcal{S}}, x, \Psi, M)$: Takes public parameters $\mathsf{params}_{\mathcal{S}}$, a public statement x, a signature Ψ, and a message M as input, returns a bit $b \in \{0, 1\}$.

3) Structure-Preserving Signatures on Equivalence Classes: SPS-EQ [21], [27] are randomizable signatures that establish relationships within message space. The equivalence relation, i.e., $\vec{M} \sim_{\mathcal{R}} \vec{N} \Leftrightarrow \exists \rho \in \mathbb{Z}_q : \vec{N} = \vec{M}^{\rho}$ for $\vec{M}, \vec{N} \in \mathbb{G}_i^{\ell} (i \in \{1, 2\})$ partitions the message space into equivalence classes. Scaling an $\mathsf{SPS\text{-}EQ}$ signature from M to N using a randomizer ρ without knowing the knowledge of the signing key maintains the class-hiding property, which ensures unlinkability between message-signature pairs of the same equivalence class.

Definition 5 ($\mathsf{SPS\text{-}EQ}$): A structure-preserving signature on equivalence classes scheme proposed by Fuchsbauer et al. [21], which is a set of $\mathcal{PPT}$ algorithms $\Pi_{\mathcal{Q}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{ChgRep}, \mathsf{Verify})$ such that the following.

1) $\mathbb{BG} \leftarrow \mathsf{Setup}(1^{\kappa})$: Takes a security parameter 1^{κ} as input and outputs a bilinear group $\mathbb{BG} = (\hat{e}, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, \widetilde{g}, g_T)$ of order q with $\lceil \log_2 q \rceil = \kappa$.
2) $(\vec{\mathsf{pk}}_{\mathcal{Q}}, \vec{\mathsf{sk}}_{\mathcal{Q}}) \leftarrow \mathsf{KeyGen}(\mathbb{BG}, \ell)$: Takes a bilinear group $\mathbb{BG}$ and a vector length $\ell \geq 2$ as input, selects $(\mathsf{sk}_i)_{i\in[\ell]} \in (\mathbb{Z}_q^*)^{\ell}$, then sets the secret key $\vec{\mathsf{sk}}_{\mathcal{Q}} = (\mathsf{sk}_i)_{i\in[\ell]}$, computes public key $\vec{\mathsf{pk}}_{\mathcal{Q}} = (\widetilde{\mathsf{pk}}_i)_{i\in[\ell]} = (\widetilde{g}^{\mathsf{sk}_i})_{i\in[\ell]}$, finally outputs a key pair $(\vec{\mathsf{pk}}_{\mathcal{Q}}, \vec{\mathsf{sk}}_{\mathcal{Q}})$.
3) $\sigma \leftarrow \mathsf{Sign}(\vec{\mathsf{sk}}_{\mathcal{Q}}, \vec{M})$: Takes a secret key $\vec{\mathsf{sk}}_{\mathcal{Q}}$ and a representative $\vec{M} = (M_i)_{i\in[\ell]} \in \mathbb{G}_1^{\ell}$ as input, chooses $y \in \mathbb{Z}_q$, returns a signature $\sigma = (\sigma_1, \sigma_2, \widetilde{\sigma}_3) = (\prod_{i\in[\ell]} (M_i^{\mathsf{sk}_i})^y, g^{(1/y)}, \widetilde{g}^{(1/y)})$ of the equivalence class $[\vec{M}]$.
4) $(\vec{M}^{\rho}, \sigma^*) \leftarrow \mathsf{ChgRep}(\vec{M}, \sigma, \rho, \vec{\mathsf{pk}}_{\mathcal{Q}})$: Takes a representative $\vec{M} \in \mathbb{G}_1^{\ell}$, a signature σ of $\vec{M}$, a value ρ, as well as a public key $\vec{\mathsf{pk}}_{\mathcal{Q}}$. Then, it picks $\phi \in \mathbb{Z}_q$ and outputs a new representative $\vec{M}^{\rho} \in \mathbb{G}_1^{\ell}$ along with a updated signature $\sigma^* = (\sigma_1^{\phi\rho}, \sigma_2^{(1/\phi)}, \widetilde{\sigma}_3^{(1/\phi)})$ of $\vec{M}^{\rho}$.
5) $b \leftarrow \mathsf{Verify}(\vec{M}, \sigma, \vec{\mathsf{pk}}_{\mathcal{Q}})$: Takes a representative $\vec{M}$, a signature σ, and a public key $\vec{\mathsf{pk}}_{\mathcal{Q}}$ as input, and checks $\prod_{i\in[\ell]} \hat{e}(M_i, \widetilde{\mathsf{pk}}_i) = \hat{e}(\sigma_1, \widetilde{\sigma}_3) \wedge \hat{e}(\sigma_2, \widetilde{g}) = \hat{e}(g, \widetilde{\sigma}_3)$, returns a bit b, where $b = 0$ states that the signature σ is invalid, and $b = 1$ states it is valid.

Definition 6 (Perfect Adaptation of Signatures): For $\ell > 1$, an $\mathsf{SPS\text{-}EQ}$ on $(\mathbb{G}_i^*)^{\ell}$ perfectly adapts signatures if for every $(\vec{\mathsf{sk}}_{\mathcal{Q}}, \vec{\mathsf{pk}}_{\mathcal{Q}}, \vec{M}, \sigma, \rho)$ where it holds that $\vec{M} \in (\mathbb{G}_i^*)^{\ell} \wedge \mathsf{Verify}(\vec{M}, \sigma, \vec{\mathsf{pk}}_{\mathcal{Q}}) = 1 \wedge \rho \in \mathbb{Z}_q^*$, the output $\mathsf{ChgRep}(\vec{M}, \sigma, \rho, \vec{\mathsf{pk}}_{\mathcal{Q}})$ s.t. $\mathsf{Verify}(\vec{M}^{\rho}, \sigma', \vec{\mathsf{pk}}_{\mathcal{Q}}) = 1$ is distributed identically to $\mathsf{Sign}(\vec{\mathsf{sk}}_{\mathcal{Q}}, \vec{M}^{\rho})$.

In other words, this definition (perfect adaptation) means that the signatures produced by ChgRep have the same distribution as the fresh signatures generated by the new representative, which yields a type of unlinkability. Due to this feature, SPS-EQ schemes ensure the hiding of renters' public keys within the car-sharing system, especially when randomizing message-signature pairs received from SP.

4) Bivariate Polynomial: A bivariate polynomial $\mathcal{F}(\cdot,\cdot)$ [28], [29], with degree d and variables $\mathcal{X}$, $\mathcal{Y}$, is defined to meet the following conditions: $\mathcal{F}(\mathcal{X},\mathcal{Y}) = \sum_{0\le i,j\le d}\theta_{ij}\mathcal{X}^i\mathcal{Y}^j \bmod q$, where θ_{ij} $(0 \le i,j \le d)$ is the polynomial coefficient which is a positive integer randomly chosen from the finite field $GF(q)$, and q is a prime number. Thus, $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{F}(\mathcal{X},\mathcal{Y})$ are all elements on $GF(q)$. When the bivariate polynomial is applied to key update and revocation [29], [30], [31], each user u_i is assigned a polynomial fragment before it is deployed in the network. The polynomial fragment of u_i is defined as $\Theta_i(\mathcal{Y}) = \mathcal{F}(u_i,\mathcal{Y})$. Each user u_i keeps $d+1$ coefficients f_j $(0 \le j \le d)$ of $\Theta_i(\mathcal{Y})$, where $f_j = \sum_{k=0}^{d}\theta_{kj}u_i^k$.

IV. Construction of PEACS

A. Overview

We leverage a Two-Tier anonymous credential model to enhance the compliance of renters' behaviors, which bolsters the security of the transaction. In this model, renters initially obtain an anonymous credential (i.e., SPS-EQ) from the SP. Subsequently, the PLs check for verification of rental permissions and overall car condition during the rental phase. Upon successful completion of this inspection, renters receive a supplementary credential (i.e., ITEQ) associated with the initial anonymous credential obtained from the SP. The scheme enhances trust in the authenticity and integrity verification of renters' requests by incorporating approval and supervision from PLs.

Our construction starts by representing an SPS-EQ signature as a renter's credential on his public key. Once a renter has passed the identity legality and driving qualification authentication by SP, SP issues a credential to the renter by performing an SPS-EQ signature on the renter's public key and computing a polynomial fragment on the renter's identity. When a renter requests for car-sharing services, he will show a valid credential in a randomized version and an SoK scheme, where the SoK is used for conditional anonymous authentication. When the renter goes to a PLs to pick up or return a sharing car, the corresponding PLs will make an ITEQ signature on the credential presented by the renter. Notably, unique randomized versions of credentials are presented for each PL, using randomizers t_p and t_r, respectively. After that, the renter updates the ITEQ signatures from the pick-up and return PLs and compresses them into a compact signature for SP. A detailed overview of the entire process can be found in Fig. 3.

The use of two PLs is integral to our proposed system as it serves to illustrate the scenario of multiparty involvement in credential presentation and validation, enhancing trust in verifying their authenticity and integrity. Our proposed scheme is tailored for station-based scenarios, where users both pick up and return vehicles at PLs. In practice, these two activities often occur at different locations. The choice to use two distinct randomized versions of anonymous credentials shown to PLs is motivated by the necessity to prevent entities (other than SP) from linking the renter's single driving record. This avoids potential inferences regarding the identity or behavioral patterns of renters.

Compared to standard SPS-EQ signatures, ITEQ signatures we proposed constitute identity-based cryptographic primitives that eliminate the need for certificate management for multiple PLs. Inspired by [32], we introduce an additional tag to mitigate the risk of replay attacks by malicious renters. For instance, a malicious renter could arbitrarily randomize a ITEQ signature previously issued from a PLs to deceive SP's verification after failing to return the vehicle on time. To prevent this, we set the tag value to the current day's timestamp, meaning the signature is signed and verified with the current date. By introducing tags and limiting the rental period to one day, replay attacks can be mitigated to some extent. Furthermore, our scheme incorporates multiparty approval to enhance trust in renter behaviors by leveraging the aggregability property of ITEQ. This property allows multiple ITEQ signatures under different identity keys and tags (representing a timestamp) on the same message (representative) to be compressed into a single signature.

B. Designed Building Blocks

In the section, we first design an SoK scheme for conditional anonymous authentication, Then, we construct an identity-based structure-preserving signature scheme with a tag on equivalence classes for PLs to further guarantee the security and reliability of car sharing transactions.

1) Signatures of Knowledge: In our PEACS, TA owns a pair of master public keys $(\mathsf{mpk}, \widetilde{\mathsf{mpk}})$ and each renter owns a long-term key pair $(\mathsf{usk}, \mathsf{upk} = g^{\mathsf{usk}})$. During the car rental phase, a legitimate renter, with a valid credential $cred^{(pre)} = ((cred_1^{(pre)}, cred_2^{(pre)}), \sigma^{(pre)}) = ((g, \mathsf{upk}), \sigma^{(pre)})$ from an SP, randomizes $cred^{(pre)}$ into a new credential $cred = ((cred_1, cred_2), \sigma)$ using a randomizer $z \in \mathbb{Z}_q^*$, where $cred_1 = g^z$, $cred_2 = \mathsf{upk}^z$ and σ is a fresh SPS-EQ signature. Then, he computes a one-time anonymous identity $nym = \mathsf{mpk}^z \cdot \mathsf{upk}$ using the same randomizer z.

Next, the renter computes the SoK Ψ to prove that: 1) I encrypted my public key upk using randomness z; 2) I know the knowledge of usk associated with the encrypted identity upk in the one-time anonymous identity nym; and 3) the encrypted identity upk in the anonymous identity nym is the one registered renter's identity credential $cred$. In essence, the first point guarantees that the anonymous identity nym can be traced to obtain a real identity upk by TA. The second point prevents the SP from forging requests on behalf of the renter. The last point prevents malicious renters from using public keys that are not been registered with the SP.

Note that, we use ElGamal encryption [33] to encrypt the real identity. Formally, $\Psi = \mathsf{SoK}[(x,w) : nym = \mathsf{mpk}^z \cdot \mathsf{upk} \wedge \mathsf{upk} = g^{\mathsf{usk}} \wedge cred_1 = g^z \wedge cred_2 = \mathsf{upk}^z](M)$, where the public statement x is $(g, \mathsf{mpk}, nym, cred_1, cred_2)$, the witness w is $(\mathsf{usk}, \mathsf{upk}, z)$. Fig. 2 illustrates the entire process of our

$$\Psi = \mathsf{SoK}[(x, w) : nym = \mathsf{mpk}^z \cdot \mathsf{upk} \wedge \mathsf{upk} = g^{\mathsf{usk}}$$
$$\wedge cred_1 = g^z \wedge cred_2 = \mathsf{upk}^z](M)$$
$$\text{where } x = (g, \mathsf{mpk}, nym, cred_1, cred_2), w = (\mathsf{usk}, \mathsf{upk}, z)$$

Prover / Signer:

- Picks blinding values $\varrho_1, \varrho_2 \in_R \mathbb{Z}_q^*$.
- Computes

$$Q_1 = \mathsf{mpk}^{\varrho_1} g^{\varrho_2}, \quad Q_2 = g^{\varrho_1}, \quad Q_3 = cred_1^{\varrho_2}.$$

- Computes the challenge

$$c = \mathcal{H}(nym, cred_1, cred_2, Q_1, Q_2, Q_3, M).$$

- Computes $s_z = \varrho_1 - z \cdot c \bmod q$, $s_{\mathsf{usk}} = \varrho_2 - \mathsf{usk} \cdot c \bmod q$.
- Sends the message M and proof $\Psi = (c, s_z, s_{\mathsf{usk}})$.

Verifier:

- Computes

$$Q_1' = \mathsf{mpk}^{s_z} g^{s_{\mathsf{usk}}} nym^c,$$
$$Q_2' = g^{s_z} {cred_1}^c, \quad Q_3' = cred_1^{s_{\mathsf{usk}}} {cred_2}^c.$$

- Accepts if this check

$$c \overset{?}{=} \mathcal{H}\left(nym, cred_1, cred_2, Q_1', Q_2', Q_3', M\right)$$

succeeds, and rejects otherwise.

Fig. 2. Details of the designed SoK scheme.

SoK scheme, where in PEACS, renters act as provers while SPs act as verifiers.

2) Identity-Based Structure-Preserving Signatures With Tag on Equivalence Classes: We introduce an identity-based SPS-EQ that supports an additional auxiliary tag $\tau \in \{0, 1\}^*$. ITEQ draws inspiration from TBEQ [32] as well as identity-based cryptographic system features, toward enriching the class of structure-preserving cryptographic schemes. Here, we clarify the algorithms of ITEQ as follows.

1) $\mathsf{params}_{\mathcal{I}} \leftarrow \mathsf{Setup}(1^\kappa)$: Given a security parameter κ as an input, it selects a bilinear group $\mathbb{BG} = (\hat{e}, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, \widetilde{g}, g_T)$. Also, it chooses four secure hash functions: a) $\mathcal{H}_1 : \mathbb{G}_2 \to \mathbb{Z}_q$; b) $\mathcal{H}_2 : \{0, 1\}^* \to \mathbb{Z}_q$; c) $\mathcal{H}_3 : \{0, 1\}^* \times \mathbb{G}_2 \to \mathbb{Z}_q$; and d) $\mathcal{H}_4 : \{0, 1\}^* \to \mathbb{G}_2$. Then, it selects a secret $\mathsf{msk} \leftarrow \mathbb{Z}_q^*$ as the master secret key and computes the master public key $\widetilde{\mathsf{mpk}} = \widetilde{g}^{\mathsf{msk}}$. Finally, it returns the public parameters $\mathsf{params}_{\mathcal{I}} = (\mathbb{BG}, \widetilde{\mathsf{mpk}}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4)$. All algorithms utilize $\mathsf{params}_{\mathcal{I}}$ as default input public parameters.
2) $(\mathsf{aid}, \mathsf{sk}_{\mathsf{ID}}) \leftarrow \mathsf{Extract}(\mathbb{BG}, \mathsf{rid}, \ell)$: Given a bilinear group $\mathbb{BG}$, a user's real identity rid, and vector length ℓ, it runs the following steps.
   a) Chooses ℓ random numbers $(s_i)_{i\in[\ell]} \in (\mathbb{Z}_q^*)^\ell$, then computes the anonymous identity of the user $\mathsf{aid} = (\mathsf{aid}_i)_{i\in[\ell+1]}$, where $(\mathsf{aid}_i)_{i\in[\ell]} = \widetilde{g}^{s_i}$ and $\mathsf{aid}_{\ell+1} = \mathsf{rid} \oplus \mathcal{H}_1(\widetilde{\mathsf{mpk}}^{s_1})$.
   b) Computes the secret key of the user $\mathsf{sk}_{\mathsf{ID}} = (\mathsf{sk}_{\mathsf{ID}_i})_{i\in[\ell]} = (s_i + \mathcal{H}_2(\mathsf{aid}_i) \cdot \mathsf{msk} \bmod \mathrm{q}_{i\in[\ell]})$.
   c) Returns $(\mathsf{aid}, \mathsf{sk}_{\mathsf{ID}})$.
3) $(\sigma, \vec{R}) \leftarrow \mathsf{Sign}(\mathsf{aid}, \mathsf{sk}_{\mathsf{ID}}, \tau, \vec{M})$: Given an anonymous identity of a user $\mathsf{aid} = (\mathsf{aid}_i)_{i\in[\ell+1]}$, secret keys $\mathsf{sk}_{\mathsf{ID}} = (\mathsf{sk}_{\mathsf{ID}_i})_{i\in[\ell]} = (s_i + \mathcal{H}_2(\mathsf{aid}_i) \cdot \mathsf{msk})_{i\in[\ell] \bmod q}$, a tag τ, and a representative $\vec{M} = (M_i)_{i\in[\ell]}$, it runs as follows.
   a) Chooses l random numbers $(r_i)_{i\in[\ell]} \in (\mathbb{Z}_q^*)^\ell$, then computes $\widetilde{R}_i = \widetilde{g}^{r_i}$.
   b) Computes $\alpha_i = \mathsf{sk}_{\mathsf{ID}_i} + \mathcal{H}_3(\mathsf{aid}_i, \widetilde{R}_i) \cdot r_i \bmod q$ for all $i \in [\ell]$.
   c) Chooses a random number $\beta \in \mathbb{Z}_q^*$, then computes $Z = \prod_{i\in[\ell]}(M_i^{\alpha_i})^\beta, Y = g^{(1/\beta)}, \tilde{Y} = \widetilde{g}^{(1/\beta)}$, and $\tilde{V} = \mathcal{H}_4(\mathsf{aid}, \tau)^{(1/\beta)}$.
   d) Returns the signature $\sigma = (Z, Y, \tilde{Y}, \tilde{V})$ and auxiliary parameters $\vec{R} = (\widetilde{R}_i)_{i\in[\ell]}$.
4) $b \leftarrow \mathsf{Verify}(\mathsf{aid}, \vec{R}, \widetilde{\mathsf{mpk}}, \vec{M}, \tau, \sigma)$: Given an anonymous identity aid of a user, auxiliary parameters $\vec{R} = (\widetilde{R}_i)_{i\in[\ell]}$, the master public key $\widetilde{\mathsf{mpk}}$, a representative $\vec{M} = (M_i)_{i\in[\ell]}$, a tag τ, and a signature $\sigma = (Z, Y, \tilde{Y}, \tilde{V})$, it runs the following steps.
   a) Checks the freshness of the timestamp T, this algorithm returns 0 if it is not and continues otherwise.
   b) Computes for all $i \in [\ell] : a_i = \mathcal{H}_2(\mathsf{aid}_i), b_i = \mathcal{H}_3(\mathsf{aid}_i, \widetilde{R}_i)$.
   c) Returns 1 if the following equations hold and 0 otherwise: $\prod_{i\in[\ell]} \hat{e}(M_i, \mathsf{aid}_i \cdot \widetilde{\mathsf{mpk}}^{a_i} \cdot \widetilde{R}_i^{b_i}) = \hat{e}(Z, \tilde{Y}) \wedge \hat{e}(Y, \widetilde{g}) = \hat{e}(g, \tilde{Y}) \wedge \hat{e}(g, \tilde{V}) = \hat{e}(Y, \mathcal{H}_4(\mathsf{aid}, \tau))$.
5) $(\vec{M}^\rho, \sigma^*) \leftarrow \mathsf{ChgRep}(\vec{M}, \sigma, \rho, \mathsf{aid}, \vec{R}, \widetilde{\mathsf{mpk}})$: Given a representative $\vec{M} = (M_i)_{i\in[\ell]}$, a signature $\sigma = (Z, Y, \tilde{Y}, \tilde{V})$, a scalar $\rho \in \mathbb{Z}_q$, an anonymous identity aid of a user, auxiliary parameters $\vec{R} = (\widetilde{R}_i)_{i\in[\ell]}$, and the master public key mpk, it picks a randomness $\psi \in_R \mathbb{Z}_q$, then returns an updated signature $\sigma^* = (Z^{\psi\rho}, Y^{(1/\psi)}, \tilde{Y}^{(1/\psi)}, \tilde{V}^{(1/\psi)})$ and a new representative $\vec{M}^\rho$.

Here, we show a variant of ITEQ scheme, which will act as a fundamental building block for PEACS scheme. Like TBEQ, the ITEQ signatures signed by different users and tags sharing the same randomness β for the same message can be compressed into a multisignature. We denote the variant of ITEQ $\Pi_{\mathcal{I}} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Sign}, \mathsf{ChgRep}, \mathsf{Verify})$ scheme by ITEQ^+ $\Pi_{\mathcal{I}}^+ = (\mathsf{Setup}, \mathsf{Extract}', \mathsf{Sign}', \mathsf{ChgRep}, \mathsf{Verify}, \mathsf{Mcomb}, \mathsf{MVerify})$. An additional step has been added to $\Pi_{\mathcal{I}}^+.\mathsf{Extract}'$ with respect to $\Pi_{\mathcal{I}}.\mathsf{Extract}$, which involves generating a key k. And the randomness β in $\Pi_{\mathcal{I}}.\mathsf{Sign}$ is generated by a secure pseudo random function $PRF(k, str)$ in $\Pi_{\mathcal{I}}^+.\mathsf{Sign}'$, where k is a key and str is a string $str \in \{0, 1\}^*$. For clarity, the algorithms for a multisignature generation and verification are as follows.

1) $\sigma_{\mathsf{mul}} \leftarrow \mathsf{Mcomb}((\sigma_j, \mathsf{tuple}_j)_{j\in[t]}, \vec{M})$: Given t signatures $(\sigma_j, \mathsf{tuple}_j)_{j\in[t]}$ on a common representative $\vec{M} = (M_i)_{i\in[\ell]}$, where $\sigma_j = (Z_j, Y_j, \tilde{Y}_j, \tilde{V}_j)$, $\mathsf{tuple}_j = (\mathsf{aid}_j, R_j, \tau)$, this algorithm returns a multisignature $\sigma_{\mathsf{mul}} = (\prod_{j\in[t]} Z_j, Y_1, \tilde{Y}_1, \prod_{j\in[t]} \tilde{V}_j)$ if all signatures are valid and $\perp$ otherwise.
2) $b \leftarrow \mathsf{MVerify}(\sigma_{\mathsf{mul}}, (\mathsf{tuple}_j)_{j\in[t]}, \vec{M})$: Given a multisignature $\sigma_{\mathsf{mul}} = (Z, Y, \tilde{Y}, \tilde{V})$, some parameters $(\mathsf{tuple}_j)_{j\in[t]} = (\mathsf{aid}_j, R_j, \tau)_{j\in[t]}$, and a representative

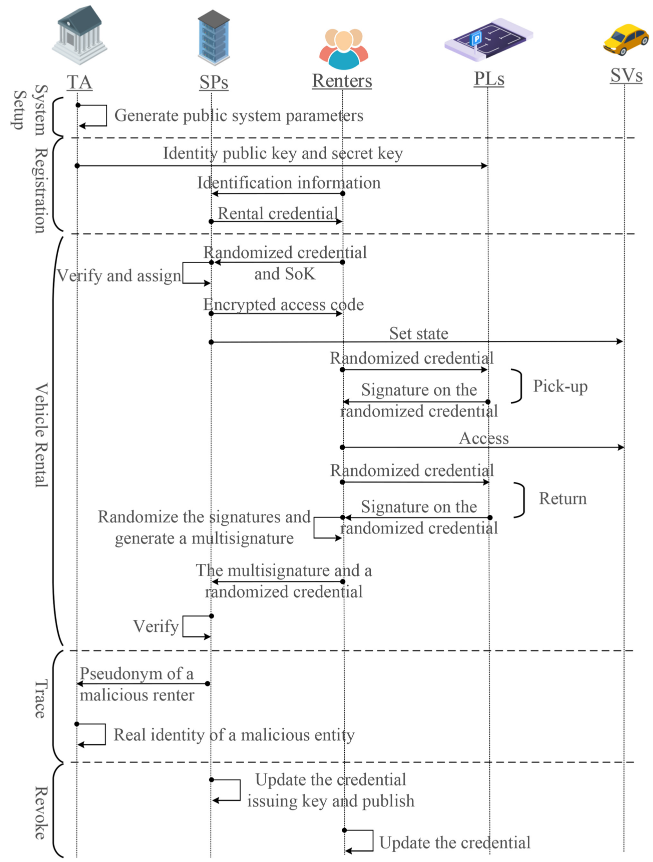
TA
SPs
Renters
PLs
SVs
System Setup
Registration
Vehicle Rental
Trace
Revoke
Generate public system parameters
Identity public key and secret key
Identification information
Rental credential
Randomized credential and SoK
Verify and assign
Encrypted access code
Set state
Randomized credential
Signature on the randomized credential
Pick-up
Access
Randomized credential
Signature on the randomized credential
Return
Randomize the signatures and generate a multisignature
The multisignature and a randomized credential
Verify
Pseudonym of a malicious renter
Real identity of a malicious entity
Update the credential issuing key and publish
Update the credential

Fig. 3. Overview of PEACS.

$\vec{M} = (M_i)_{i\in[\ell]}$, it returns 1 if the equations $\prod_{i\in[\ell]} \hat{e}(M_i, \prod_{j\in[t]}(\mathsf{aid}_{j,i} \cdot \widetilde{\mathsf{mpk}}^{a_{j,i}} \cdot \tilde{R}_{j,i}^{b_{j,i}})) = \hat{e}(Z, \tilde{Y}) \wedge \hat{e}(Y, \tilde{g}) = \hat{e}(g, \tilde{Y}) \wedge \hat{e}(g, \tilde{V}) = \hat{e}(Y, \prod_{j\in[t]} \mathcal{H}_4(\mathsf{aid}_j, \tau))$ hold and 0 otherwise.

C. Detailed Construction of PEACS Scheme

The designed PEACS is constructed from the following building blocks: the SPS-EQ signatures scheme $\Pi_{\mathcal{Q}} = (\mathsf{Setup}, \mathsf{KeyGen}, \mathsf{Sign}, \mathsf{ChgRep}, \mathsf{Verify})$, the designed SoK scheme $\Pi_{\mathcal{S}} = (\mathsf{Setup}, \mathsf{Sign}, \mathsf{Vrf})$, and the designed ITEQ^{+} signatures scheme $\Pi_{\mathcal{I}}^{+} = (\mathsf{Setup}, \mathsf{Extract}, \mathsf{Sign}, \mathsf{ChgRep}, \mathsf{Verify}, \mathsf{Mcomb}, \mathsf{MVerify})$. In a bit more detail, PEACS is composed of the following polynomial-time algorithms as is shown in Fig. 3. For notational convenience, we describe a single SP in the protocol. It is apparent to see that our scheme can be easily scaled to multiple SPs.

1) System Setup: TA generates the public system parameters by executing the algorithms $\Pi_{\mathcal{Q}}.\mathsf{Setup}$, $\Pi_{\mathcal{S}}.\mathsf{Setup}$, and $\Pi_{\mathcal{I}}^{+}.\mathsf{Setup}$. Moreover, it needs to compute $\mathsf{mpk} = g^{\mathsf{msk}} \in \mathbb{G}_1$ using the master secret key msk and sets the system master public key as $\mathsf{MPK} = (\mathsf{mpk}, \widetilde{\mathsf{mpk}})$ where $\widetilde{\mathsf{mpk}}$ is obtained from $\Pi_{\mathcal{I}}^{+}.\mathsf{Setup}$. Thereafter, TA runs $\Pi_{\mathcal{Q}}.\mathsf{KeyGen}(\mathbb{BG}, 2)$ to generate SP's credential issuing key $(\vec{\mathsf{sk}}_{\mathcal{Q}} = (\mathsf{sk}_1, \mathsf{sk}_2), \vec{\mathsf{pk}}_{\mathcal{Q}} = (\tilde{\mathsf{pk}}_1, \tilde{\mathsf{pk}}_2))$ and sends the key pair to SP via a private channel, e.g., SSL-TLS [34]

$$\mathsf{params} = \Big\{\vec{\mathsf{pk}}_{\mathcal{Q}}, \mathbb{BG} = (\hat{e}, q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, g, \tilde{g}, g_T)$$
$$\mathsf{MPK} = (\mathsf{mpk}, \widetilde{\mathsf{mpk}}), \mathcal{H}, \mathcal{H}_1, \mathcal{H}_2, \mathcal{H}_3, \mathcal{H}_4\Big\}.$$

TA publishes the public system parameters params and locally stores msk. All the following algorithms take params as implicit input public parameters.

SP generates a d-degree bivariate polynomial $\mathcal{F}(\mathcal{X}, \mathcal{Y})$ for the revocation process, where $\mathcal{X}$, $\mathcal{Y}$ are variables and d means the maximum number of revoked renters.

2) Renter Registration: The very basic idea is that a renter when wanting to get a car rental service registers himself with SP. Renters need to provide SP with several necessary documents about identification for a check, e.g., its identity, public key, valid driving license, and insurance. For the purpose of ease, we assume a renter named Alice with an identity uid_i interacts with SP as follows.

1) Alice generates her key pair $(\mathsf{upk}, \mathsf{usk})$, where $\mathsf{usk} \in_R \mathbb{Z}_q^*$ and $\mathsf{upk} = g^{\mathsf{usk}}$. After that, Alice chooses $r \in_R \mathbb{Z}_q^*$ and sends a registration request containing a signature $\sigma = (c, s)$, a driving license, and insurance to SP, where $c = \mathcal{H}(g^r, \mathsf{upk})$, $s = r + c \cdot usk \bmod q$.
2) SP checks if the signature σ is valid via verifying the equation $c \overset{?}{=} \mathcal{H}(g^s \mathsf{upk}^{-c}, \mathsf{upk})$. If σ, the driving license, and the insurance are all valid, SP runs $\Pi_{\mathcal{Q}}.\mathsf{Sign}(\vec{\mathsf{sk}}_{\mathcal{Q}}, (g, \mathsf{upk}))$ to generate a rental credential $\sigma^{(pre)} = (\sigma_1^{(pre)}, \sigma_2^{(pre)}, \tilde{\sigma}_3^{(pre)})$, then computes a polynomial fragment $\Theta_i(\mathcal{Y}) = \mathcal{F}(\mathsf{uid}_i, \mathcal{Y})$. Eventually, SP sends $(\sigma^{(pre)}, \Theta_i(\mathcal{Y}))$ for Alice via a private channel and saves $(\mathsf{uid}_i, \mathsf{upk})$ into the registration list.
3) Upon receiving $(\sigma^{(pre)}, \Theta_i(\mathcal{Y}))$ from SP, Alice checks $\sigma^{(pre)}$ by running the algorithm $\Pi_{\mathcal{Q}}.\mathsf{Verify}((g, \mathsf{upk}), \sigma^{(pre)}, \vec{\mathsf{pk}}_{\mathcal{Q}})$. Alice then sets $cred^{(pre)} = ((cred_1^{(pre)}, cred_2^{(pre)}), \sigma^{(pre)}) = ((g, \mathsf{upk}), \sigma^{(pre)})$ as her credential if $\sigma^{(pre)}$ is valid. Otherwise, the registration fails.

3) Parking Lot Registration: Our scheme focuses on station-based vehicular sharing. Hence, we require that each PLs be registered with TA. TA executes $((\mathsf{aid}, \mathsf{sk}_{\mathsf{ID}}), k) \leftarrow \Pi_{\mathcal{I}}^{+}.\mathsf{Extract}(\mathbb{BG}, \mathsf{rid}, \ell = 2)$ for a PL via a private channel, where rid is the real identification (e.g., location and the SP it belongs to) of the PL, $\mathsf{aid} = (\mathsf{aid}_i)_{i\in[\ell+1]}$ is the anonymous identity whose corresponding secret key is $\mathsf{sk}_{\mathsf{ID}} = (\mathsf{sk}_{\mathsf{ID}_i})_{i\in[\ell]}$, k is the key of PRF. We assume that PLs belonging to the same SP use a shared PRF key.

4) Vehicle Rental: A renter with an issued credential named Alice who is looking for a car can request vehicular sharing services to the corresponding SP. The following interactive steps are performed with an SP, a PL for pick-up $\mathsf{PL_p}$, and a PL for return $\mathsf{PL_r}$ to complete the rental phase. Note that our basic idea is to use the current date as the tag τ in the ITEQ scheme. For example, the tag on February 24, 2023, is defined as “20230224.”

1) Alice executes the following steps.
   a) Selects two random numbers $z, r \in \mathbb{Z}_q^*$.
   b) Switches her credential $cred^{(pre)} = ((g, \mathsf{upk}), \sigma^{(pre)})$ to a random representative $((g^z, \mathsf{upk}^z), \sigma)$ by running $\Pi_{\mathcal{Q}}.\mathsf{ChgRep}((g, \mathsf{upk}), \sigma^{(pre)}, z, \vec{\mathsf{pk}}_{\mathcal{Q}})$,

   and sets the updated credential $cred = ((cred_1, cred_2), \sigma)$, where $cred_1 = g^z, cred_2 = \mathsf{upk}^z$, and σ is the updated signature of $\sigma^{(pre)}$.

   c) Computes $nym = \mathsf{mpk}^z \cdot \mathsf{upk}$ for accountability, and sets nym as her pseudonym.

   d) Generates a rental request Req, including information on an appropriate vehicle and the rental period.

   e) Computes a proof $\Psi = \mathsf{SoK}[(\mathsf{usk}, \mathsf{upk}, z) : nym = \mathsf{mpk}^z \cdot \mathsf{upk} \wedge \mathsf{upk} = g^{\mathsf{usk}} \wedge cred_1 = g^z \wedge cred_2 = \mathsf{upk}^z](Req)$ by invoking $\Pi_{\mathcal{S}}.\mathsf{Sign}(\mathsf{params}, x, w, Req)$ where the public statement is $x = (g, \mathsf{mpk}, nym, cred_1, cred_2)$, the witness is $w = (\mathsf{usk}, \mathsf{upk}, z)$.

   f) Sends $(cred, nym, \Psi, Req)$ to an SP.

2) Upon SP receiving $(cred, nym, \Psi, Req)$ from Alice, he executes the following steps.

   a) Checks $cred = ((cred_1, cred_2), \sigma)$ to see if the credential is valid by running $\Pi_{\mathcal{Q}}.\mathsf{Verify}(cred, \vec{\mathsf{pk}}_{\mathcal{Q}})$. If not, he rejects this request.

   b) Validates Ψ by running $\Pi_{\mathcal{S}}.\mathsf{Vrf}(\mathsf{params}, x, \Psi, Req)$. If Ψ is not valid, he rejects this request.

   c) If $cred$ and Ψ are both valid, SP assigns an appropriate vehicle for Alice according to the rental request Req. Otherwise, he rejects it.

   d) Encrypts the access code of the SV using $cred_1 = g^z$ as $code^*$ and adds a tuple consisting of the credential $cred$, anonymous identity nym of Alice, and the rental period to the maintained rental list $\mathbb{L}$.

   e) Sends $code^*$ to Alice and starts timing.

3) Alice picks a randomness $t_p \in \mathbb{Z}_q^*$, gets a randomized credential $cred^{(p)}$ by $((g^{t_p}, \mathsf{upk}^{t_p}), \sigma^{(p)}) \leftarrow \Pi_{\mathcal{Q}}.\mathsf{ChgRep}((g, \mathsf{upk}), \sigma^{(pre)}, t_p, \vec{\mathsf{pk}}_{\mathcal{Q}})$, then goes to the designated location to pick up the vehicle by showing the credential $cred^{(p)}$. Then, she decrypts the $code^*$ into the access code by using the secret key z of $cred_1$. Subsequently, the PL $\mathsf{PL_p}$ proceeds as follows.

   a) Checks the credential $cred^{(p)}$ to see if it is valid by running $\Pi_{\mathcal{Q}}.\mathsf{Verify}(cred^{(p)}, \vec{\mathsf{pk}}_{\mathcal{Q}})$. If not, $\mathsf{PL_p}$ rejects this request.

   b) Generates a signature σ_p on $cred^{(p)}$ by invoking $(\sigma_p, \vec{R}_p) \leftarrow \Pi_{\mathcal{I}}^{+}.\mathsf{Sign}(\mathsf{aid}_p, \mathsf{sk}_{\mathsf{ID}_p}, \tau, (g^{t_p}, \mathsf{upk}^{t_p}); CN)$, where $\mathsf{aid}_p = (\mathsf{aid}_{p,i})_{i \in [3]}$ and $\mathsf{sk}_{\mathsf{ID}_p} = (\mathsf{sk}_{\mathsf{ID}_{p,i}})_{i \in [2]}$ is the anonymous identity and secret key of $\mathsf{PL_p}$, τ is the current date, and CN is the car number used in PRF as $PRF(k, \tau || CN)$.

   c) Sends $(\sigma_p, \vec{R}_p, \mathsf{aid}_p)$ to Alice.

   When Alice exceeds the rental period and does not return the SV, SP would collect evidence for TA to punish Alice, i.e., expose her real identity.

4) When Alice wants to return the vehicle, she obtains a randomized credential $cred^{(r)}$ by $((g^{t_r}, \mathsf{upk}^{t_r}), \sigma^{(r)}) \leftarrow \Pi_{\mathcal{Q}}.\mathsf{ChgRep}\ ((g, \mathsf{upk}), \sigma^{(pre)}, t_r, \vec{\mathsf{pk}}_{\mathcal{Q}})$ with a random number $t_r \in \mathbb{Z}_q^*$. Then, she uploads the credential $cred^{(r)}$ and several photographs of the SV for the PL $\mathsf{PL_r}$ to check the vehicle condition. $\mathsf{PL_r}$ proceeds as follows after receiving the returning request from Alice.

   a) Checks the credential $cred^{(r)}$ to see if it is valid by running $\Pi_{\mathcal{Q}}.\mathsf{Verify}(cred^{(r)}, \vec{\mathsf{pk}}_{\mathcal{Q}})$. If not, $\mathsf{PL_r}$ rejects this request.

   b) Checks if the SV is properly returned.

   c) Generates a signature σ_r on $cred^{(r)}$ by invoking $(\sigma_r, \vec{R}_r) \leftarrow \Pi_{\mathcal{I}}^{+}.\mathsf{Sign}(\mathsf{aid}_r, \mathsf{sk}_{\mathsf{ID}_r}, \tau, (g^{t_r}, \mathsf{upk}^{t_r}); CN)$, where $\mathsf{aid}_r = (\mathsf{aid}_{r,i})_{i \in [3]}$ and $\mathsf{sk}_{\mathsf{ID}_r} = (\mathsf{sk}_{\mathsf{ID}_{r,i}})_{i \in [2]}$ is the anonymous identity and secret key of $\mathsf{PL_r}$, τ is the current date, and CN is the car number used in PRF as $PRF(k, \tau || CN)$. If valid, continues. Otherwise, goes to Trace algorithm.

   d) Sends $(\sigma_r, \vec{R}_r, \mathsf{aid}_r)$ to Alice.

5) After receiving $(\sigma_p, \vec{R}_p, \mathsf{aid}_p)$ and $(\sigma_r, \vec{R}_r, \mathsf{aid}_r)$ from $\mathsf{PL_p}$ and $\mathsf{PL_r}$, respectively, Alice follows these steps.

   a) Invokes $\Pi_{\mathcal{I}}^{+}.\mathsf{Verify}(\mathsf{aid}_p, \vec{R}_p, \widetilde{\mathsf{mpk}}, cred^{(p)}, \tau, \sigma_p)$ and $\Pi_{\mathcal{I}}^{+}.\mathsf{Verify}(\mathsf{aid}_r, \vec{R}_r, \widetilde{\mathsf{mpk}}, cred^{(r)}, \tau, \sigma_r)$, respectively, to verify the two signatures σ_p and σ_r. If both are valid, continues. Otherwise, goes to Trace algorithm.

   b) Changes the representation of σ_p into σ_p' by $((g^z, \mathsf{upk}^z), \sigma_p') \leftarrow \Pi_{\mathcal{I}}^{+}.\mathsf{ChgRep}((g^{t_p}, \mathsf{upk}^{t_p}), \sigma_p, t_p^{-1}z, \mathsf{aid}_p, \vec{R}_p, \widetilde{\mathsf{mpk}})$, and changes σ_r to σ_r' by $((g^z, \mathsf{upk}^z), \sigma_r') \leftarrow \Pi_{\mathcal{I}}^{+}.\mathsf{ChgRep}((g^{t_r}, \mathsf{upk}^{t_r}), \sigma_r, t_r^{-1}z, \mathsf{aid}_r, \vec{R}_r, \widetilde{\mathsf{mpk}})$ using the same randomness ψ so that the signed message are (g^z, upk^z), which is consistent with $cred$, and will be used by the SP as an index to locate the rental records.

   c) Invokes $\Pi_{\mathcal{I}}^{+}.\mathsf{Mcomb}(\sigma_p', \mathsf{aid}_p, \vec{R}_p, \sigma_r', \mathsf{aid}_r, \vec{R}_r, (g^z, \mathsf{upk}^z))$ to generate a multisignature σ_{mul}.

   d) Sends $(\mathsf{aid}_p, \vec{R}_p, \mathsf{aid}_r, \vec{R}_r, \sigma_{\mathsf{mul}}, cred)$ to SP.

6) Once SP receiving $(\mathsf{aid}_p, \vec{R}_p, \mathsf{aid}_r, \vec{R}_r, \sigma_{\mathsf{mul}}, cred)$ from Alice, he executes the following steps.

   a) Locates Alice's rental information in the rental list $\mathbb{L}$ according to $cred$.

   b) Checks if the SV is in good condition. If so, continues. Otherwise, goes to Trace algorithm.

   c) Checks the validity of σ_{mul} by invoking $\Pi_{\mathcal{I}}^{+}.\mathsf{MVerify}(\sigma_{\mathsf{mul}}, \mathsf{aid}_p, \vec{R}_p, \mathsf{aid}_r, \vec{R}_r, (g^z, \mathsf{upk}^z))$. If valid, continues. Else, goes to Trace algorithm.

   d) Stops the timer, then updates the SV's state, and settlements the rental fees according to the elapsed time.

Remarks: The use of distinct randomized versions of anonymous credentials is crucial to address a potential privacy vulnerability. If the same version of the anonymous credential were used for both picking up and returning, an adversary could exploit the timing of the PLs issuing an anonymous credential to deduce the renter's driving schedule, thereby compromising privacy. To mitigate this risk, the system employs two versions of the anonymous credential, denoted as (t_p, t_r). During the return process, the renter rerandomizes the anonymous credential back to the initially registered version, denoted as (z), at the SP for settlement purposes. By severing the correlation between picking up and returning times, the system effectively conceals the renter's driving schedule.

Note that anonymous payment for car rental fees is not within the scope of this work. However, existing anonymous payment schemes, such as BlockMaze [35], may be useful.

5) Trace: TA can not only expose the identity of misbehaved renters but also his traveling track. Once TA receives a complaint request, it confirms the validity of the evidence and computes $nym \cdot cred_1^{-\mathsf{msk}} = (\mathsf{mpk}^z \cdot \mathsf{upk}) \cdot (g^z)^{-\mathsf{msk}} = \mathsf{upk}$ with its private key msk to recover the identity of the misbehaved renter. Moreover, if necessary, TA can reveal his traveling track as follows. Using the master secret key msk, TA computes $\mathsf{aid}_{p,1}^{\mathsf{msk}} = (\widetilde{g}^{s_{p,1}})^{\mathsf{msk}} = (\widetilde{g}^{\mathsf{msk}})^{s_{p,1}} = \widetilde{\mathsf{mpk}}^{s_{p,1}}$, then computes $\mathsf{rid}_p = \mathsf{aid}_{p,\ell+1} \oplus \mathcal{H}_1(\mathsf{aid}_{p,1}^{\mathsf{msk}})$ to reveal the real identity of the PL $\mathsf{PL_p}$, similarly for $\mathsf{PL_r}$.

6) Renter Revocation: SP regularly changes its credential issuing key $\vec{\mathsf{pk}}_{\mathcal{Q}}$ to revoke malicious renters while forcing the honest renters to regularly update their credentials. We denote the revoked renters list as $RL = \{\mathsf{uid}_{i_1}, \ldots, \mathsf{uid}_{i_n}\}$, which is publicly available to anyone.

1) SP updates its credential issuing key by following.
   a) Picks a secret randomness $\lambda \in \mathbb{Z}_q^*$, and then sets $\vec{\mathsf{pk}}'_{\mathcal{Q}} = (\widetilde{\mathsf{pk}}_1^{\lambda}, \widetilde{\mathsf{pk}}_2^{\lambda})$ as the new credential issuing key. After that, computes $o = \mathcal{H}_2(\vec{\mathsf{pk}}'_{\mathcal{Q}}, T)$, where T is the current timestamp.
   b) Computes the following polynomials based on the identities of revoked renters:
   $$\Delta(\mathcal{X}) = \left(\mathcal{X} - \mathsf{uid}_{i_1}\right) \cdots \left(\mathcal{X} - \mathsf{uid}_{i_n}\right)$$
   $$\Gamma(\mathcal{X}) = \lambda \Delta(\mathcal{X}) + \mathcal{F}(\mathcal{X}, o)$$
   where $\Delta(\mathcal{X})$ is called the revocation polynomial, $\Gamma(\mathcal{X})$ is dubbed the broadcast polynomial.
   c) Generates a signature σ_{ϖ} on $\mathcal{H}_2(\Gamma(\mathcal{X}), o, T)$ using any secure signature scheme, e.g., Boneh–Lynn–Shacham signature [36], by employing sk_1.
   d) Broadcasts the message tuple $(\vec{\mathsf{pk}}'_{\mathcal{Q}}, \Gamma(\mathcal{X}), \sigma_{\varpi}, T)$.
2) If Alice uid_i with a rental credential $cred^{(pre)} = ((g, \mathsf{upk}), \sigma^{(pre)} = (\sigma_1^{(pre)}, \sigma_2^{(pre)}, \widetilde{\sigma}_3^{(pre)}))$ is not revoked, she can recover the secret λ to update her credential.
   a) Checks if $T - T'$ is acceptable, where T' is the timestamp when Alice receives the broadcast message. If so, computes $o = \mathcal{H}_2(\vec{\mathsf{pk}}'_{\mathcal{Q}}, T)$, and verifies the signature σ_{ϖ} using the old credential issuing key $\widetilde{\mathsf{pk}}_1$. If σ_{ϖ} is valid, accepts the tuple $(\vec{\mathsf{pk}}'_{\mathcal{Q}}, \Gamma(\mathcal{X}), \sigma_{\varpi}, T)$.
   b) Calculates $\lambda = [(\Gamma(\mathsf{uid}_i) - \Theta_i(o))/(\Delta(\mathsf{uid}_i))]$, where the polynomial fragment $\Theta_i(\mathcal{Y}) = \mathcal{F}(\mathsf{uid}_i, \mathcal{Y})$ is secretly obtained from SP when Alice is registered.
   c) Updates her old credential $\sigma^{(pre)} = (\sigma_1^{(pre)}, \sigma_2^{(pre)}, \widetilde{\sigma}_3^{(pre)})$ to a new credential $\sigma^{(pre)*} = ((\sigma_1^{(pre)})^{\lambda}, \sigma_2^{(pre)}, \widetilde{\sigma}_3^{(pre)})$.

It is easy to verify that the equations $\hat{e}(g, \widetilde{\mathsf{pk}}'_1)\hat{e}(\mathsf{upk}, \widetilde{\mathsf{pk}}'_2) = \hat{e}((\sigma_1^{(pre)})^{\lambda}, \widetilde{\sigma}_3) \wedge \hat{e}(\sigma_2^{(pre)}, \widetilde{g}) = \hat{e}(g, \widetilde{\sigma}_3^{(pre)})$ hold. Assuming that a renter with identifier uid_{i_r} has been revoked, it is impossible for him to compute the secret λ because $\Delta(\mathsf{uid}_{i_r}) = 0$, which means that he cannot update his credential.

V. Security Analysis

Theorem 1: The SoK designed above is a NIZK argument with perfect completeness, special soundness, and honest-verifier zero-knowledge in the random oracle model.

Proof (Sketch) Completeness: It can be readily verified that if the prover submits all values as instructed, then the following equations are upheld:

$$\begin{aligned} Q'_1 &= g^{s_z} {cred_1}^c = g^{\varrho_1 - z \cdot c} {cred_1}^c = g^{\varrho_1} {cred_1}^{-c} {cred_1}^c = Q_1 \\ Q'_2 &= \mathsf{mpk}^{s_z} g^{s_{\mathsf{usk}}} nym^c = \mathsf{mpk}^{\varrho_1 - z \cdot c} g^{\varrho_2 - \mathsf{usk} \cdot c} nym^c \\ &= \mathsf{mpk}^{\varrho_1} g^{\varrho_2} \left(\mathsf{mpk}^z g^{\mathsf{usk}}\right)^{-c} nym^c = \mathsf{mpk}^{\varrho_1} g^{\varrho_2} = Q_2 \\ Q'_3 &= cred_1^{s_{\mathsf{usk}}} {cred_2}^c = cred_1^{\varrho_2 - \mathsf{usk} \cdot c} {cred_2}^c \\ &= g^{z(\varrho_2 - \mathsf{usk} \cdot c)} g^{(\mathsf{usk} \cdot z) \ \cdot c} = {cred_1}^{\varrho_2} = Q_3 \end{aligned}$$

which ensures that the last condition is satisfied.

Special Soundness: Assuming the DL assumption, the soundness can be proved under the random oracle model. Specifically, we suppose that a $\mathcal{PPT}$ prover $\mathcal{P}^*$ generates an accepted transcript $(M, c, s_z, s_{\mathsf{usk}})$. After that, we can construct a knowledge extractor $\texttt{Ext}$ using standard rewinding techniques to obtain another valid transcript $(M, c', s'_z, s'_{\mathsf{usk}})$, where $c \neq c'$. Then, the extractor $\mathcal{E}$ can extract the witness $z = (s_z - s'_z)(c' - c)^{-1}$, $\mathsf{usk} = (s_{\mathsf{usk}} - s'_{\mathsf{usk}})(c' - c)^{-1}$. It is worth noting that all of the equations above are performed modulo q. Since $c, c' \in \mathbb{Z}_q$, then $(c' - c)^{-1} \bmod q$ can be calculated efficiently.

Honest-Verifier Zero Knowledge: We describe such a simulator $\texttt{Sim}$ with a statement $x = (g, \mathsf{mpk}, nym, cred_1, cred_2)$ to prove this property by the standard Fiat–Shamir heuristic method. To simulate the conversation with honest verifiers, $\texttt{Sim}$ randomly chooses $\mathfrak{c}, \mathfrak{s}_z, \mathfrak{s}_{\mathsf{usk}} \in \mathbb{Z}_q$, then computes $\mathfrak{Q}_1 = g^{\mathfrak{s}_z} {cred_1}^{\mathfrak{c}}$, $\mathfrak{Q}_2 = \mathsf{mpk}^{\mathfrak{s}_z} g^{\mathfrak{s}_{\mathsf{usk}}} nym^{\mathfrak{c}}$, $\mathfrak{Q}_3 = cred_1^{\mathfrak{s}_{\mathsf{usk}}} {cred_2}^{\mathfrak{c}}$. $\texttt{Sim}$ sets the hash function $\mathcal{H}$ as a random oracle $\mathcal{RO}$ and responses $\mathcal{H}(nym, cred_1, cred_2, \mathfrak{Q}_1, \mathfrak{Q}_2, \mathfrak{Q}_3, M)$ with $\mathfrak{c}$. The transcript output by $\texttt{Sim}$ denotes $(M, \mathfrak{Q}_1, \mathfrak{Q}_2, \mathfrak{Q}_3, \mathfrak{c}, \mathfrak{s}_z, \mathfrak{s}_{\mathsf{usk}})$. Upon inspection, it can be verified that the transcript satisfies all checks of the verifier.

It is yet to be demonstrated that the distribution of transcripts generated by $\texttt{Sim}$ and those generated by the honest prover and verifier are equivalent. To name a few, a choice for either of $\mathfrak{s}_z$ or ϱ_1, with z and $\mathfrak{c}$ fixed, determines the other where $\mathfrak{s}_z = \varrho_1 - \mathfrak{c} \cdot z$. That is, selecting one uniformly at random will give the other being uniformly selected as well. Therefore, $\mathfrak{s}_z$ and $\mathfrak{Q}_1$ follow an identical distribution as that of a real transcript, similarly for $\mathfrak{s}_{\mathsf{usk}}$. As argued above, the transcript output is indistinguishable from a real transcript by any particular prover. That is, for any possible transcript, the probability of its production by both the honest prover and the simulator $\texttt{Sim}$ is the same, which concludes the proof. ■

Theorem 2: ITEQ scheme proposed is correct.

Proof (Sketch): Due to $\widetilde{\mathsf{mpk}} = \widetilde{g}^{\mathsf{msk}}$, $\mathsf{aid} = ((\mathsf{aid}_i)_{i \in [\ell]}, \mathsf{aid}_{\ell+1}) = ((\widetilde{g}^{s_i})_{i \in [\ell]}, \mathsf{rid} \oplus \mathcal{H}_1(\widetilde{\mathsf{mpk}}^{s_1}))$, $\mathsf{sk_{ID}} = (\mathsf{sk}_{\mathsf{ID}_i})_{i \in [\ell]} = (s_i + \mathcal{H}_2(\mathsf{aid}_i) \cdot \mathsf{msk})_{i \in [\ell]}$, $\alpha_i = \mathsf{sk}_{\mathsf{ID}_i} + \mathcal{H}_3(\mathsf{aid}_i, \widetilde{R}_i) \cdot r_i$, $\widetilde{R}_i = \widetilde{g}^{r_i}$, $a_i = \mathcal{H}_2(\mathsf{aid}_i)$, $b_i = \mathcal{H}_3(\mathsf{aid}_i, \widetilde{R}_i)$, $Z = \prod_{i \in [\ell]} (M_i^{\alpha_i})^{\beta}$, $Y = g^{(1/\beta)}$, $\tilde{Y} = \widetilde{g}^{(1/\beta)}$, and $\tilde{V} = \mathcal{H}_4(\mathsf{aid}, \ \tau)^{(1/\beta)}$, we can obtain that

$$\begin{aligned} \hat{e}\big(Z, \tilde{Y}\big) &= \hat{e}\left(\prod_{i \in [\ell]} (M_i^{\alpha_i})^{\beta}, \widetilde{g}^{\frac{1}{\beta}}\right) = \hat{e}\left(\prod_{i \in [\ell]} M_i^{\alpha_i}, \widetilde{g}\right)^{\beta \cdot \frac{1}{\beta}} \\ &= \prod_{i \in [l]} \hat{e}\Big(M_i, \widetilde{g}^{(\mathsf{sk}_{\mathsf{ID}_i} + \mathcal{H}_3(\mathsf{aid}_i, \widetilde{R}_i) \cdot r_i)}\Big) \end{aligned}$$

$$= \prod_{i\in[l]} \hat{e}\Big(M_i, \widetilde{g}^{s_i}(\widetilde{g}^{\mathsf{msk}})^{\mathcal{H}_2(\mathsf{aid}_i)}(\widetilde{g}^{r_i})^{\mathcal{H}_3(\mathsf{aid}_i,\widetilde{R}_i)}\Big)$$

$$= \prod_{i\in[l]} \hat{e}\Big(M_i, \mathsf{aid}_i \cdot \widetilde{\mathsf{mpk}}^{a_i} \cdot \widetilde{R}_i^{b_i}\Big). \quad (1)$$

It also holds that $\hat{e}(Y, \widetilde{g}) = \hat{e}(g^{(1/\beta)}, \widetilde{g}) = \hat{e}(g, \widetilde{g}^{(1/\beta)}) = \hat{e}(g, \tilde{Y})$ and $\hat{e}(g, \tilde{V}) = \hat{e}(g, \mathcal{H}_4(\mathsf{aid}, \tau)^{(1/\beta)}) = \hat{e}(g^{(1/\beta)}, \mathcal{H}_4(\mathsf{aid}, \tau)) = \hat{e}(Y, \mathcal{H}_4(\mathsf{aid}, \tau))$.

Therefore, the correctness of ITEQ holds.

Assuming that a multisignature $\sigma_{\mathsf{mul}} = (\prod_{j\in[t]} Z_j, Y_1, \tilde{Y}_1, \prod_{j\in[t]} \tilde{V}_j)$ is generated by t ITEQ signatures $(Z_j, Y_j, \tilde{Y}_j, \tilde{V}_j)_{j\in[t]}$. The correctness of ITEQ$^+$ holds because

$$\hat{e}\left(\prod_{j\in[t]} Z_j, \tilde{Y}_1\right) = \hat{e}\left(\prod_{j\in[t]}(\prod_{i\in[\ell]}(M_i^{\alpha_{j,i}})^{\beta_1}), \widetilde{g}^{\frac{1}{\beta_1}}\right)$$

$$= \hat{e}\left(\prod_{j\in[t]}(\prod_{i\in[\ell]}(M_i^{\alpha_{j,i}})), \widetilde{g}\right)^{\beta_1\cdot\frac{1}{\beta_1}}$$

$$= \prod_{i\in[\ell]} \hat{e}\left(M_i, \prod_{j\in[t]} \widetilde{g}^{\mathsf{sk}_{\mathsf{ID}_{j,i}}+\mathcal{H}_3(\mathsf{aid}_{j,i},\widetilde{R}_{j,i})\cdot r_{j,i}}\right)$$

$$= \prod_{i\in[\ell]} \hat{e}\left(M_i, \prod_{j\in[t]}(\widetilde{g}^{s_{j,i}}(\widetilde{g}^{\mathsf{msk}})^{\mathcal{H}_2(\mathsf{aid}_{j,i})}(\widetilde{g}^{r_{j,i}})^{\mathcal{H}_3(\mathsf{aid}_{j,i},\widetilde{R}_{j,i})})\right)$$

$$= \prod_{i\in[\ell]} \hat{e}\left(M_i, \prod_{j\in[t]}(\mathsf{aid}_{j,i} \cdot \widetilde{\mathsf{mpk}}^{a_{j,i}} \cdot \widetilde{R}_{j,i}^{b_{j,i}})\right) \quad (2)$$

$$\hat{e}\left(Y_1, \prod_{j\in[t]} \mathcal{H}_4(\mathsf{aid}_j, \tau)\right)$$

$$= \hat{e}\left(g, \prod_{j\in[t]} \mathcal{H}_4(\mathsf{aid}_j, \tau)^{\frac{1}{\beta}}\right) = \hat{e}\left(g, \prod_{j\in[t]} \tilde{V}_j\right). \quad (3)$$

Hence, we complete the proof. ■

Theorem 3: ITEQ scheme provides EUF-CMA secure for Type-3 bilinear groups in the generic group model.

proof (Sketch): We strictly follow the proof of the SPS-EQ scheme in [21]. The group elements the adversary has seen from q queries, besides g and $\tilde{g}$, are $(Z_j, Y_j)_{j\in[q]}$ in $\mathbb{G}_1$, and $(\tilde{Y}_j, \tilde{V}_j)_{j\in[q]}$, mpk as well as $(\mathsf{aid}_i, \tilde{R}_i)_{i\in[\ell]}$ in $\mathbb{G}_2$. We denote $\mathsf{aid}_i, \widetilde{\mathsf{mpk}}^{\mathcal{H}_2(\mathsf{aid}_i)}, \tilde{R}_i^{\mathcal{H}_3(\mathsf{aid}_i,\widetilde{R}_i)}$ as $\tilde{g}^{x_i}, \tilde{g}^{u_i}, \tilde{g}^{t_i}$, respectively. By taking their DLs of the forgery $(Z^*, Y^*, \tilde{Y}^*, \tilde{V}^*)$ to the bases g and $\tilde{g}$, respectively, we can get the following equations:

$$z^* = \pi_z + \sum_{j\in[q]} \rho_{z,j} z_j + \sum_{j\in[q]} \psi_{z,j}\frac{1}{y_j}$$

$$y^* = \pi_y + \sum_{j\in[q]} \rho_{y,j} z_j + \sum_{j\in[q]} \psi_{y,j}\frac{1}{y_j}$$

$$\tilde{y}^* = \pi_{\tilde{y}} + \sum_{i\in[\ell]} \chi_{\tilde{y},i} x_i + \sum_{i\in[\ell]} \omega_{\tilde{y},i} u_i + \sum_{i\in[\ell]} \xi_{\tilde{y},i} t_i$$
$$+ \sum_{i\in[p]} \theta_{\tilde{y},i} h_i + \sum_{j\in[q]} \nu_{\tilde{y},j} v_j + \sum_{j\in[q]} \psi_{\tilde{y},j}\frac{1}{y_j}$$

$$\tilde{v}^* = \pi_{\tilde{v}} + \sum_{i\in[\ell]} \chi_{\tilde{v},i} x_i + \sum_{i\in[\ell]} \omega_{\tilde{v},i} u_i + \sum_{i\in[\ell]} \xi_{\tilde{v},i} t_i$$
$$+ \sum_{i\in[p]} \theta_{\tilde{v},i} h_i + \sum_{j\in[q]} \nu_{\tilde{v},j} v_j + \sum_{j\in[q]} \psi_{\tilde{v},j}\frac{1}{y_j}$$

$$m_i^* = \pi_{m^*,i} + \sum_{j\in[q]} \rho_{m^*,i,j} z_j + \sum_{j\in[q]} \psi_{m^*,i,j}\frac{1}{y_j}$$

$$m_{j,i} = \pi_{m,j,i} + \sum_{k\in[j-1]} \rho_{m,j,i,k} z_k + \sum_{k\in[j-1]} \psi_{m,j,i,k}\frac{1}{y_k}. \quad (4)$$

According to the verification equations of ITEQ, we take DLs to the base $e(g, \tilde{g})$, and know that

$$\sum_{i\in[\ell]} m_i^*(x_i + u_i + t_i) = z^*\tilde{y}^* \quad (5a)$$

$$y^* = \tilde{y}^* \quad (5b)$$

$$\tilde{v}^* = y^*\tilde{h}^*. \quad (5c)$$

Taking into account that by Claim 1 and Corollary 1 from the proof in [21], we start by investigating (5b) and comparing the coefficients implies that $\pi_y = \pi_{\tilde{y}}$, and $\chi_{\tilde{y},i} = \omega_{\tilde{y},i} = \xi_{\tilde{y},i} = 0$ for all $i \in [\ell]$, $\theta_{\tilde{y},i} = 0$ for all $i \in [k]$, $\rho_{y,j} = 0$, $\nu_{\tilde{y},j} = 0$, $\psi_{y,j} = \psi_{\tilde{y},j}$ for all $j \in [q]$. It simplifies (5b) to the following:

$$y^* = \pi_y + \sum_{j\in[q]} \psi_{y,j}\frac{1}{y_j}. \quad (6)$$

The investigation of (5c) is identical to the proof of [32, Lemma 3.7]. Hence, we omit it here. We now plug (6) into (5a), which leads to

$$\sum_{i\in[\ell]}\left(\pi_{m^*,i} + \sum_{j\in[q]} \rho_{m^*,i,j} z_j + \sum_{j\in[q]} \psi_{m^*,i,j}\frac{1}{y_j}\right)(x_i + u_i + t_i)$$
$$= \left(\pi_z + \sum_{j\in[q]} \rho_{z,j} z_j + \sum_{j\in[q]} \psi_{z,j}\frac{1}{y_j}\right)$$
$$\left(\pi_y + \sum_{j\in[q]} \psi_{y,j}\frac{1}{y_j}\right). \quad (7)$$

By expanding the right-hand side of (7) and equating coefficients, we get $\pi_z\pi_y = 0, \pi_y\psi_{z,j} + \pi_z\psi_{y,j} = 0$, $\psi_{z,j}\psi_{y,j} = 0$ for all $j \in [q]$. Moreover, we have $\rho_{z,j}\pi_y z_n = 0$ for all $j \in [q]$ by [21, Corollary 1]. Now the right-hand side of (7) has been simplified as follows: $\sum_{j\in[q]}\sum_{k\in[q]} \rho_{z,j}\psi_{y,k}(1/y_k)z_j$. Next, Claim 1 in [21] shows that each term of z_j contains an equal number of x's and y's (u's and t's, respectively) in the numerator. Thus, on the left-hand side of (7) the number of x's (u's and t's, respectively) is 1 more than that of y's. Following [21], we obtain that $\rho_{z,j}\psi_{y,k} = 0$ for all $j \in [q]\backslash k$, and $\rho_{z,n}\psi_{y,n} \neq 0$ for some $!n \in [q]$. Therefore, the right-hand side of (7) can be simplified further: $\rho_{z,n}\psi_{y,n}(1/y_n)z_n$.

We replace z_n and $m_{n,i}$ by their definitions, respectively, i.e., $z_n = \sum_{k\in[\ell]} m_{n,k}x_k$, $m_{n,i} = (\pi_{m,n,i} + \sum_{k\in[n-1]} \rho_{m,n,i,k}z_k + \sum_{k\in[n-1]} \psi_{m,n,i,k}(1/y_k))$ such that

$$\rho_{z,n}\psi_{y,n}\sum_{i\in[\ell]}\left(\pi_{m,n,i} + \sum_{k\in[n-1]} \rho_{m,n,i,k} z_k + \sum_{k\in[n-1]} \psi_{m,n,i,k}\frac{1}{y_k}\right)$$
$$(x_i + u_i + t_i). \quad (8)$$

By comparing coefficients with the left-side hand of (7), let $\mu = \rho_{z,n}\psi_{y,n}$, we have $\pi_{m^*,i} = \mu\pi_{m,n,i}$, $\rho_{m^*,i,j} = \mu\rho_{m,n,i,k}$, and $\psi_{m^*,i,j} = \mu\psi_{m,n,i,k}$. Thus, we know that the forgery is just a combination of messages from previous queries, so we prove the EUF-CMA security. It is worth noting that the analysis of the probability of an adversary accidentally producing an existential forgery aligns with [21]. ■

Theorem 4: ITEQ scheme has perfect adaption.

proof (Sketch): Assume that $\vec{M} = (M_i)_{i\in[\ell]} \in (\mathbb{G}_1^*)^\ell$, $\mathsf{aid} = ((\mathsf{aid}_i)_{i\in[\ell]}, \mathsf{aid}_{\ell+1}) = ((\tilde{g}^{s_i})_{i\in[\ell]}, \mathsf{rid} \oplus \mathcal{H}_1(\widetilde{\mathsf{mpk}}^{s_1})) \in (\mathbb{G}_2^*)^{\ell+1}$, $\widetilde{\mathsf{mpk}} = \tilde{g}^{\mathsf{msk}} \in \mathbb{G}_2^*$, $\vec{R} = (\tilde{g}^{r_i})_{i\in[\ell]} \in (\mathbb{G}_2^*)^\ell$, $\mathcal{H}_1 : \mathbb{G}_2 \to \mathbb{Z}_q$, $\mathcal{H}_2 : \{0,1\}^* \to \mathbb{Z}_q$, $\mathcal{H}_3 : \{0,1\}^* \times \mathbb{G}_2 \to \mathbb{Z}_q$, $\mathcal{H}_4 : \{0,1\}^* \to \mathbb{G}_2$, and $\tau \in \{0,1\}^*$. A signature $(Z, Y, \tilde{Y}, \tilde{V}) \in \mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{G}_2$ satisfying $\mathsf{Verify}(\mathsf{aid}, \vec{R}, \widetilde{\mathsf{mpk}}, \vec{M}, \tau, (Z, Y, \tilde{Y}, \tilde{V})) = 1$ has the form $(\prod(M_i^{s_i+\mathcal{H}_2(\mathsf{aid}_i)\cdot\mathsf{msk}+\mathcal{H}_3(\mathsf{aid}_i,\tilde{R}_i)\cdot r_i})^\beta, g^{(1/\beta)}, \tilde{g}^{(1/\beta)}, \mathcal{H}_4(\mathsf{aid}, \tau)^{(1/\beta)})$ for some $\beta \in \mathbb{Z}_q^*$. $\mathsf{ChgRep}(\vec{M}, (Z, Y, \tilde{Y}, \tilde{V}), \rho, \mathsf{aid}, \vec{R}, \widetilde{\mathsf{mpk}})$ outputs $\sigma = \sigma = (\prod(M_i^{s_i+\mathcal{H}_2(\mathsf{aid}_i)\cdot\mathsf{msk}+\mathcal{H}_3(\mathsf{aid}_i,\tilde{R}_i)\cdot r_i})^{\rho\beta\phi}, g^{\frac{1}{\beta\phi}}, \tilde{g}^{\frac{1}{\beta\phi}}, \mathcal{H}_4(\mathsf{aid}, \tau)^{\frac{1}{\beta\phi}})$ for some $\phi \in \mathbb{Z}_q^*$, which randomly lies in a uniform distribution $\mathbb{G}_1 \times \mathbb{G}_1^* \times \mathbb{G}_2^* \times \mathbb{G}_2$ s.t. $\mathsf{Verify}(\mathsf{aid}, \vec{R}, \widetilde{\mathsf{mpk}}, \vec{M}^\rho, \tau, \sigma) = 1$. ■

A. Security Analysis of PEACS

In this part, we describe that PEACS can ensure the security and privacy properties mentioned in Section II-C.

Security:

1) *Entity Authentication:* Due to the unforgeability of SPS-EQ scheme [21], any adversary cannot forge a valid SPS-EQ signature as his credential. Also, owing to the completeness and soundness of our designed SoK scheme, no $\mathcal{PPT}$ adversary can produce a convincing SoK without a legal credential. Moreover, due to ITEQ unforgeability, malicious renters cannot forge the ITEQ signature of a PLs and pass the verification by SPs without returning shared cars or damaging them. Thereby, SPs can confirm the authenticity of a received service request and a correct returning request.
2) *Accountability:* The accountability property mainly consists of traceability and revocability. Since our proposed SoK is sound, we can conclude that upk is indeed hidden in *nym* and can be decrypted by TA correctly. That is, the SoK signatures always can be traced back to a renter who made it. Hence, the traceability is guaranteed in PEACS. The revocability property is satisfied by updating the SP's credential issuing key $\vec{\mathsf{pk}}_Q = (\tilde{\mathsf{pk}}_1, \tilde{\mathsf{pk}}_2)$ to $\vec{\mathsf{pk}}'_Q = (\tilde{\mathsf{pk}}_1^\lambda, \tilde{\mathsf{pk}}_2^\lambda)$. A misbehaved renter with uid^* in the revoked list *RL* cannot obtain λ by computing $[(\Gamma(\mathsf{uid}^*) - \Theta_i(o))/(\Delta(\mathsf{uid}^*))]$, that is because $\Delta(\mathsf{uid}^*) = 0$. Therefore, the renter uid^* cannot update his own credential, and thus cannot generate a valid SoK for car rental services.
3) *Nonframeability:* Breaking the nonframeability of PEACS denotes that has the capability to create a valid SPS-EQ signature as well as a valid SoK, which contradicts the following facts that SPS-EQ signature scheme is EUF-CMA from [21] and our SoK is soundness under DL assumption. Thus, a malicious adversary cannot frame an honest renter who did not sign it to prevent being traced.

Privacy:

1) *Anonymity:* The anonymity property is strengthened in four ways: a) the randomized credential is used to hide the renter's real credential; b) the membership public key upk, which is also declared in a valid credential, is encrypted by TA's master public key using ElGamal encryption scheme; c) the well-designed SoK scheme satisfies zero-knowledge; and d) the unlinkability of our PEACS further enhance the anonymity property. That is, we can reduce the anonymity property to the class-hiding of SPS-EQ, IND-CPA security of ElGamal encryption scheme, and the zero-knowledge of our SoK scheme. The class-hiding property of SPS-EQ allows making sure that renters can obtain fresh credentials that are distinguishable from the original ones under the DDH assumption. If any $\mathcal{PPT}$ adversary can successfully break the anonymity of PEACS, it is contradictory with DDH assumption in $\mathbb{G}_1$ and DL assumption.
2) *Unlinkability:* This requirement can be enhanced from two aspects: a) a credential presented by a renter each time he requests the vehicle rental service is a randomized version of the original credential and b) ITEQ signatures obtained by renters from the PLs are randomized and then sent to the SP for verification. Namely, we can reduce the unlinkability property to the perfect adaptation of SPS-EQ and ITEQ. Under the perfect adaption of SPS-EQ, we have that multiple vehicle rental service records made by the same renter using the same credential cannot be linked by an adversary (including semi-honest SPs). The perfect adaptation of ITEQ indicates that the three versions of the credential presented by the renter, namely, the t_p randomized version $((g^{t_p}, \mathsf{upk}^{t_p}), \sigma^{(p)})$ presented to the pick-up parking station, the t_r randomized version $((g^{t_r}, \mathsf{upk}^{t_r}), \sigma^{(r)})$ presented to the return parking station, and the original z version $((g^z, \mathsf{upk}^z), \sigma)$, are identically distributed. By breaking the link between picking up and returning times, the system successfully hides the renter's driving schedule. Thus, in a rental transaction, adversaries are unable to link a renter with a particular trajectory, enhancing privacy protection.
3) *Location Privacy:* From the unlinkability, we have that adversaries are unable to determine a specific tracking to a renter. Additionally, each PLs obtains an anonymous identity from TA at registration, thus external adversaries are unable to track the trajectory of renters. The location privacy is effectively protected as the real location information remains concealed from the public.

VI. Evaluation and Discussion

A. Theoretical Analysis

For a better illustration of the computational and communicational costs of PEACS, Table II first gives the main

TABLE II
EXECUTION TIME OF KEY CRYPTOGRAPHIC OPERATIONS AND THE DATA SIZE OF ELEMENTS

Notations	Descriptions	Time (ms)	Notations	Descriptions	Size (bytes)
T_{exp_1}	Time of an exponentiation in the group $\mathbb{G}_1$	0.167	$\lvert\mathbb{Z}_q\rvert$	Size of a scalar value in $\lvert\mathbb{Z}_q\rvert$	39
T_{exp_2}	Time of an exponentiation in the group $\mathbb{G}_2$	0.325	$\lvert\mathbb{G}_1\rvert$	Size of a group element in $\lvert\mathbb{G}_1\rvert$	58
T_{bp}	Time of a bilinear pairing	1.152	$\lvert\mathbb{G}_2\rvert$	Size of a group element in $\lvert\mathbb{G}_2\rvert$	115
T_{exp_T}	Time of an exponentiation in the group $\mathbb{G}_T$	0.461	$\lvert\mathbb{G}_T\rvert$	Size of a group element in $\lvert\mathbb{G}_T\rvert$	684
T_{hp_1}	Time of a MapToPoint $\mathbb{G}_1$ hash operation	0.205	$\lvert k\rvert$	Size of a secret key of PRF	32
T_{hp_2}	Time of a MapToPoint $\mathbb{G}_2$ hash operation	0.538	$\lvert T\rvert$	Size of a timestamp	8
T_{EE}	Time of a ElGamal encryption	0.334	-	-	-
T_{ED}	Time of a ElGamal decryption	0.167	-	-	-

TABLE III
COMPLEXITY FOR EACH PHASES OF OUR PEACS SCHEME

Phase	Entity	Computation cost	Time (ms)	Communication cost	Size (bytes)
Renter Registration	SP	$3T_{exp_1} + 2T_{exp_2}$	1.151	$2\lvert\mathbb{Z}_q\rvert$	78
	Renter	$4T_{exp_1} + 5T_{bp}$	6.428	$2\lvert\mathbb{G}_1\rvert + \lvert\mathbb{G}_2\rvert + \lvert\mathbb{Z}_q\rvert$	270
Parking lot Registration	TA	$3T_{exp_2}$	0.975	$2\lvert\mathbb{Z}_q\rvert + 3\lvert\mathbb{G}_2\rvert + \lvert k\rvert$	455
Vehicle rental	SP	$7T_{exp_1} + 4T_{exp_2} + 12T_{bp} + T_{hp_1}$	16.498	$2\lvert\mathbb{G}_1\rvert$	116
	Renter	$18T_{exp_1} + 13T_{exp_2} + 14T_{bp} + 2T_{hp_1}$	23.769	$21\lvert\mathbb{G}_1\rvert + 16\lvert\mathbb{G}_2\rvert + 3\lvert\mathbb{Z}_q\rvert + \lvert Req\rvert$	3175+$\lvert Req\rvert$
	PL	$6T_{exp_1} + 8T_{exp_2} + 10T_{bp} + 2T_{hp_2}$	16.198	$4\lvert\mathbb{G}_1\rvert + 18\lvert\mathbb{G}_2\rvert$	2070
Trace	TA	T_{exp_1}	0.167	$2\lvert\mathbb{G}_1\rvert$	116
Renter Revocation	SP	$2T_{exp_1} + 2T_{exp_2}$	0.984	$\lvert\mathbb{Z}_q\rvert + \lvert\mathbb{G}_1\rvert + 2\lvert\mathbb{G}_2\rvert + \lvert T\rvert$	335
	Renter	$2T_{exp_1} + 2T_{bp}$	2.638	-	-

notations with their descriptions. In theoretical analysis, we focus on the most time-consuming cryptographic operations, i.e., exponentiation, bilinear pairing, and hash-to-point operation.

Above all, we describe the two designed building blocks, i.e., ITEQ and SoK. As the computational cost of ITEQ scheme is slightly lower but almost equivalent to that of ITEQ$^+$ scheme, we primarily describe ITEQ scheme here. As for ITEQ, the computation costs of $\Pi_{\mathcal{I}}$.Sign, $\Pi_{\mathcal{I}}$.Verify, and $\Pi_{\mathcal{I}}$.ChgRep are $(\ell+1)T_{exp_1}+(\ell+2)T_{exp_2}+T_{hp}$, $2\ell T_{exp_2}+(\ell+5)T_{bp}+T_{hp}$, and $2T_{exp_1}+2T_{exp_2}$, respectively, where ℓ is the length of a message vector. The size of an ITEQ signature is $2|\mathbb{G}_1|+2|\mathbb{G}_2|$. As for the well-designed SoK, the computation complexity of $\Pi_{\mathcal{S}}$.Sign and $\Pi_{\mathcal{S}}$.Vrf are both $7T_{exp_1}$, and the communication cost is $3|\mathbb{G}_1|+3|\mathbb{Z}_q|$.

Furthermore, we analyze the asymptotic efficiency of different phases in PEACS in terms of computation and communication overhead. Precisely, ℓ is set to 2 in our PEACS. Based on SPS-EQ, ITEQ, and SoK schemes, the communication and computational complexity of each stage in our scheme are summarized in Table III. We can see that the vehicle rental is the major computational and communicational burden, which is mainly composed of the following algorithms: $\Pi_{\mathcal{Q}}$.ChgRep, $\Pi_{\mathcal{Q}}$.Verify, $\Pi_{\mathcal{Q}}$.ChgRep, $\Pi_{\mathcal{S}}$.Sign, $\Pi_{\mathcal{S}}$.Vrf, $\Pi_{\mathcal{I}}$.Sign, $\Pi_{\mathcal{I}}$.Verify, $\Pi_{\mathcal{I}}$.ChgRep, $\Pi_{\mathcal{I}}$.Mcomb, $\Pi_{\mathcal{I}}$.MVerify.

B. Implementation Results

Benchmarks: To demonstrate the efficiency and feasibility of our scheme, our measurements have been performed on a personal computer equipped with Intel Core i7-10700 CPU at 2.90 GHz, 16.0-GB RAM, and 64-bit Ubuntu 18.04 operation system via utilizing the Relic Library (ver. 0.6.0) [37]. To guarantee around 128-bit security level, we employ the popular pairing-friendly 455-bit Barreto-Lynn-Scott (BLS) curve with embedding degree 12 which provides efficient Type-3 bilinear groups. This setting is the same as that of Emura et al. [38]. We utilized SHA-256 as the implementation of the employed pseudo-random function and for hashing. In our experiments, the TA, the SP, the renter, and the PLs are located on the same computer, thereby network delays are not taken into account.

The benchmarks of the key time-consuming cryptographic operations in our environment are given in Table II. We obtain the execution time of each phase from the above settings, as shown in Table III, from which we can get that the actual operation time of each entity in each phase is within 6.5 ms and the communication cost is at most 455 bytes except for the vehicle rental phase. From the previous theoretical analysis, we know that the vehicle rental phase is the most expensive in the whole system. Even so, SP, renters, and PLs only need to take about 16 ms, 23 ms, and 16 ms in computational overhead, which is acceptable in practical use.

Comparison: Renters in the vehicle rental phase of PEACS use group-signature-based approaches to submit service requests. In our scheme, the group signature-like method consists of the well-designed SoK scheme and underlying SPS-EQ scheme. Hence, to further emphasize the advantages of our PEACS, we provide a brief comparison with three existing group-signature-based service-providing systems, i.e., Shen et al. [19], Lu et al. [20] and DAPA [8]. Since the validation servers in DAPA are distributed, to guarantee fairness in the comparison phase, we set the number of validation servers of DAPA to 1 and that of revoked users to 20 and 30, respectively. We evaluate three main operations, that is, signature generation, verification, and identity tracing for misbehaved renters.

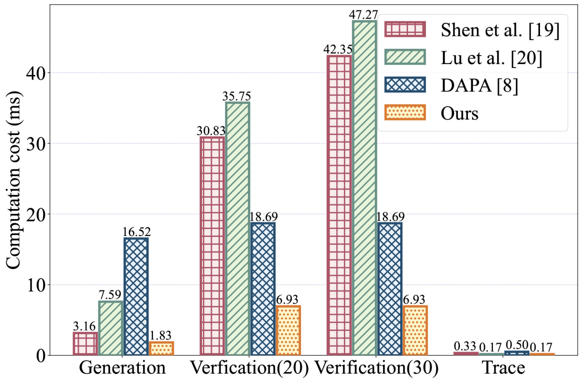

Fig. 4. Comparison of computational costs.

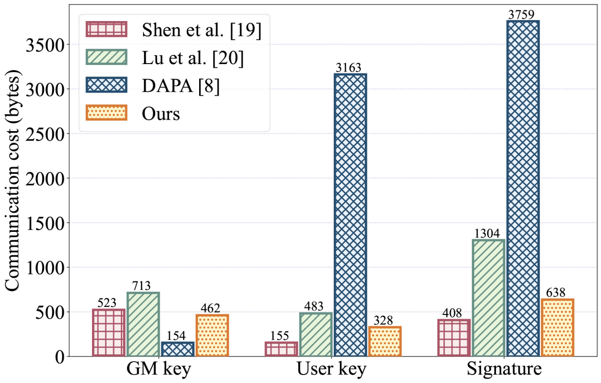

Fig. 5. Comparison of communicational costs.

We first compare the computation overhead in Fig. 4, from which we can conclude that our PEACS is the most efficient in the verification and identity tracing of group signatures. The verification in Shen et al. and Lu et al. both need to check revoked users sequentially, thus the verification time overhead is proportional to the number of revoked users. DAPA utilizes an accumulator technique, and our PEACS scheme employs a bivariate polynomial, so the verification time of both schemes is independent of the number of revoked users. It is worth mentioning that the computational overhead of each operation of our PEACS is lowest compared to the other three schemes. By comparing to DAPA which is most similar to our scheme, our PEACS can save the generation time about 88.92%, verification time about 62.92%, and identity tracing operation time about 66.00%, respectively.

Fig. 5 depicts the comparison of communicational costs. Regarding the size of the group manager (GM)'s keys, our PEACS is larger than DAPA (i.e., 462 bytes versus 154 bytes). With regard to the size of group users' keys and group signatures, Shen et al.'s scheme is shorter than ours (i.e., 155 bytes versus 328 bytes, 408 bytes versus 638 bytes). Compared with DAPA, our scheme has achieved approximately 89.63% and 83.03% reductions in communication costs for users' keys and signatures, respectively. Moreover, in our PEACS, the parking stations need to provide an ITEQ signature after checking the status of the sharing car, which strengthens the reliability of the whole car sharing system. The online interaction with the parking stations by the SP is reduced.

Unlike DAPA for the station-based model, PEACS distributes car-sharing legality checks to each station, enhancing transaction credibility, security, and deterring malicious actions. It is worth noting that, in contrast to other approaches from the above analysis, our PEACS can achieve a better tradeoff between anonymity, unlinkability, accountability, and efficiency in real-world car sharing applications.

VII. Conclusion

In this work, we present an efficient PEACS, named privacy-enhancing and accountable car sharing system, from designed cryptographic tools, the SoK scheme, and an identity-based structure-preserving signatures scheme. We not only prove the security of PEACS but also provide extensive simulation and qualitative comparison with related works. As indicated in our theoretical proof and experimental evaluations, PEACS enjoys a considerable performance advantage in computation and communication costs, which is of highly practical relevance but also of theoretical interest for deployment in station-based car sharing systems. As part of our future work, we will focus on the design of more advanced cryptographic primitives to be better compatible with real-world car sharing scenarios.

References

[1] J. Seo and S. Lee, "Who gives up a private car for a car-sharing service? An empirical case study of incheon city, South Korea," *Int. J. Sustain. Transp.*, vol. 16, no. 10, pp. 875–886, 2022.

[2] "Car-sharing - worldwide." Accessed: Jul. 15, 2023. [Online]. Available: https://www.statista.com/outlook/mmo/shared-mobility/shared-rides/car-sharing

[3] S. A. Shaheen and A. P. Cohen, "Carsharing and personal vehicle services: Worldwide market developments and emerging trends," *Int. J. Sustain. Transp.*, vol. 7, no. 1, pp. 5–34, 2013.

[4] M. Naphade, G. Banavar, C. Harrison, J. Paraszczak, and R. Morris, "Smarter cities and their innovation challenges," *Computer*, vol. 44, no. 6, pp. 32–39, Jun. 2011.

[5] E. W. Martin and S. A. Shaheen, "Greenhouse gas emission impacts of carsharing in North America," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 4, pp. 1074–1086, Dec. 2011.

[6] M. Enev, A. Takakuwa, K. Koscher, and T. Kohno, "Automobile driver fingerprinting," *Proc. Priv. Enhanc. Technol.*, no. 1, pp. 34–51, 2016.

[7] L. Zhu, M. Li, Z. Zhang, and Z. Qin, "ASAP: An anonymous smart-parking and payment scheme in vehicular networks," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 4, pp. 703–715, Jul./Aug. 2020.

[8] C. Huang, R. Lu, J. Ni, and X. Shen, "DAPA: A decentralized, accountable, and privacy-preserving architecture for car sharing services," *IEEE Trans. Veh. Technol.*, vol. 69, no. 5, pp. 4869–4882, May 2020.

[9] I. Symeonidis, D. Rotaru, M. A. Mustafa, B. Mennink, B. Preneel, and P. Papadimitratos, "HERMES: Scalable, secure, and privacy-enhancing vehicular sharing-access system," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 129–151, Jan. 2022.

[10] (Eur. Parliam., Strasbourg, France). *Council of the EU Final Compromised Resolution: General Data Protection Regulation*. (2015). [Online]. Available: http://www.europarl.europa.eu

[11] I. Symeonidis, M. A. Mustafa, and B. Preneel, "Keyless car sharing system: A security and privacy analysis," in *Proc. IEEE Int. Smart Cities Conf. (ISC2)*, 2016, pp. 1–7.

[12] I. Symeonidis, A. Aly, M. A. Mustafa, B. Mennink, S. Dhooghe, and B. Preneel, "SePCAR: A secure and privacy-enhancing protocol for car access provision," in *Proc. 22nd Eur. Symp. Res. Comput. Secur.-ESORICS*, 2017, pp. 475–493.

[13] (Indra Sistemas Inf. Technol. Co., Madrid, Spain). *SECREDAS: Product Security for Cross Domain Reliable Dependable Automated Systems*. Accessed: Jul. 15, 2023. [Online]. Available: http://cordis.europa.eu/project/id/783119

[14] M. Akash, I. Symeonidis, M. A. Mustafa, B. Preneel, and R. Zhang, "SC2Share: Smart contract for secure car sharing," in *Proc. Int. Conf. Inf. Syst. Secur. Priv.*, 2019, pp. 1–9.

[15] A. Dmitrienko and C. Plappert, "Secure free-floating car sharing for offline cars," in *Proc. 7th ACM Conf. Data Appl. Secur. Priv.*, 2017, pp. 349–360.

[16] A. Boualouache, S.-M. Senouci, and S. Moussaoui, "PRIVANET: An efficient pseudonym changing and management framework for vehicular ad-hoc networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 8, pp. 3209–3218, Aug. 2020.

[17] S. Haider, D. Gao, R. Ali, A. Hussain, and M. T. Ikram, "A privacy conserves pseudonym acquisition scheme in vehicular communication systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15536–15545, Sep. 2022.

[18] Z. Yang, W. Wang, Y. Huang, and X. Li, "Privacy-preserving public auditing scheme for data confidentiality and accountability in cloud storage," *Chin. J. Electron.*, vol. 28, no. 1, pp. 179–187, 2019.

[19] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and traceable group data sharing in cloud computing," *IEEE Trans. Inf. Forensics Security*, vol. 13, pp. 912–925, 2018.

[20] J. Lu, J. Shen, P. Vijayakumar, and B. B. Gupta, "Blockchain-based secure data storage protocol for sensors in the Industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5422–5431, Aug. 2022.

[21] G. Fuchsbauer, C. Hanser, and D. Slamanig, "Structure-preserving signatures on equivalence classes and constant-size anonymous credentials," *J. Cryptol.*, vol. 32, no. 2, pp. 498–546, 2019.

[22] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discr. Appl. Math.*, vol. 156, no. 16, pp. 3113–3121, 2008.

[23] M. Blum, P. Feldman, and S. Micali, "Non-interactive zero-knowledge and its applications," in *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. New York, NY, USA: Assoc. Comput. Mach., 2019, pp. 329–349.

[24] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Proc. Conf. Theory Appl. Cryptogr. Techn.*, 1986, pp. 186–194.

[25] I. Damgård, "On Σ-protocols," Lecture Notes CPT 2010, Aarhus Univ., Aarhus, Denmark, 2002.

[26] M. Chase and A. Lysyanskaya, "On signatures of knowledge," in *Proc. Annu. Int. Cryptol. Conf.*, 2006, pp. 78–96.

[27] C. Hanser and D. Slamanig, "Structure-preserving signatures on equivalence classes and their application to anonymous credentials," in *Proc. 20th Int. Conf. Theory Appl. Cryptol. Inf. Secur. Adv. Cryptol.–ASIACRYPT*, 2014, pp. 491–511.

[28] C. Blundo, A. D. Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly-secure key distribution for dynamic conferences," in *Proc. Annu. Int. Cryptol. Conf.*, 2001, pp. 471–486.

[29] R. Jiang, R. Lu, J. Luo, C. Lai, and X. Shen, "Efficient self-healing group key management with dynamic revocation and collusion resistance for SCADA in smart grid," *Secur. Commun. Netw.*, vol. 8, no. 6, pp. 1026–1039, 2015.

[30] Z. Ma, J. Zhang, Y. Guo, Y. Liu, X. Liu, and W. He, "An efficient decentralized key management mechanism for VANET with blockchain," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 5836–5849, Jun. 2020.

[31] Y. Wang, X. Wang, H.-N. Dai, X. Zhang, and M. Imran, "A data reporting protocol with revocable anonymous authentication for edge-assisted intelligent transport systems," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7835–7847, Jun. 2023.

[32] L. Hanzlik and D. Slamanig, "With a little help from my friends: Constructing practical anonymous credentials," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 2004–2023.

[33] Y. Tsiounis and M. Yung, "On the security of ElGamal based encryption," in *Proc. Int. Workshop Public Key Cryptogr.*, 1998, pp. 117–134.

[34] E. Rescorla, "The transport layer security (TLS) protocol version 1.3," Internet Eng. Task Force, RFC 8446, 2018.

[35] Z. Guan, Z. Wan, Y. Yang, Y. Zhou, and B. Huang, "Blockmaze: An efficient privacy-preserving account-model blockchain based on zk-SNARKS," *IEEE Trans. Dependable Secure Comput.*, vol. 19, no. 3, pp. 1446–1463, May/Jun. 2022.

[36] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proc. 7th Int. Conf. Theory Appl. Cryptol. Inf. Secur. Adv. Cryptol.—ASIACRYPT*, 2001, pp. 514–532.

[37] D. F. Aranha, C. P. L. Gouvêa, T. Markmann, R. S. Wahby, and K. Liao. "RELIC is an efficient LIbrary for cryptography." Accessed: Sep. 6, 2021. [Online]. Available: https://github.com/relic-toolkit/relic

[38] K. Emura, T. Hayashi, and A. Ishida, "Group signatures with time-bound keys revisited: A new model, an efficient construction, and its implementation," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 2, pp. 292–305, Mar./Apr. 2020.

Yulin Liu received the bachelor's degree in information security from the College of Mathematics and Computer Science, Fuzhou University, Fuzhou, China, in 2021. She is currently pursuing the Ph.D. degree with the Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China.

Her research interests mainly include information security and applied cryptography.

Debiao He (Member, IEEE) received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2009.

He is currently a Professor with the School of Cyber Science and Engineering, Wuhan University. He has authored or coauthored more than 100 research papers in refereed international journals and conferences, such as the IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING, IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, and Usenix Security Symposium. His work has been cited more than 10 000 times at Google Scholar. His main research interests include cryptography and information security, in particular cryptographic protocols.

Prof. He was the recipient of the 2018 IEEE SYSTEMS JOURNAL Best Paper Award and the 2019 *IET Information Security* Best Paper Award. He is in the editorial board of several international journals, such as the *ACM Distributed Ledger Technologies: Research and Practice*, *Frontiers of Computer Science*, and IEEE TRANSACTIONS ON COMPUTERS.

Zijian Bao received the M.S. degree in computer application technology from the School of Computer Science and Engineering, Northeastern University, Shenyang, China, in 2019. He is currently pursuing the Ph.D. degree with the Key Laboratory of Aerospace Information Security and Trusted Computing Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China.

His research interests include cryptographic protocols.

Min Luo received the Ph.D. degree in computer science from Wuhan University, Wuhan, China, in 2003.

He is currently a Professor with the school of Cyber Science and Engineering, Wuhan University. He has authored or coauthored more than 50 research papers in refereed international journals and conferences, such as IEEE SYMPOSIUM ON SECURITY AND PRIVACY and IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY. His research interests include cryptography and information security, in particular blockchain security.

Cong Peng received the Ph.D. degree in applied mathematics from the School of Mathematics and Statistics, Wuhan University, Wuhan, China, in 2021.

He is currently an Associate Professor with the School of Cyber Science and Engineering, Wuhan University. His research interests mainly include applied cryptography and data security.