DAPA: A Decentralized, Accountable, and Privacy-Preserving Architecture for Car Sharing Services

Cheng Huang, *Student Member, IEEE*, Rongxing Lu, *Senior Member, IEEE*, Jianbing Ni, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

***Abstract*—Car sharing offers a flexible peer-to-peer or station based car rental service to customers. On one hand, it requires customers to expose identifications (e.g., valid driving licenses) to car sharing service providers (CSSPs) for accountability, i.e., the driving qualification of customers can be verified and misbehaving customers can be traced by CSSPs. On the other hand, privacy concerns arise when customers identities are exposed as honest-but-curious CSSPs may secretly extract customers privacy information by linking their car rental records to their identities. To resolve this contradiction, we propose a decentralized, accountable, and privacy-preserving architecture for car sharing services, named DAPA. In specific, to overcome the limitation of the single point of failure, multiple dynamic validation servers are employed to substitute a single trusted third-party authority and assist in building decentralized trust for customers. In addition, to protect customers' privacy and achieve accountability simultaneously under the decentralized architecture, a new privacy-preserving identity management (PPIM) scheme is introduced as a basic module for DAPA. Customers' identities are protected in a distributed and dynamic manner but publicly verified based on a well-designed zero-knowledge proof protocol. Only the misbehaving customers' identities can be recovered by a majority of validation servers using adaptive verifiable secret sharing/redistribution techniques. Detailed security analysis shows that DAPA can minimize privacy breaches and guarantee the accountability. Performance evaluations via extensive simulations demonstrate that DAPA is efficient in terms of computational costs and communication overheads.**

***Index Terms*—Car sharing, decentralization, accountability, privacy preservation, identity management.**

I. Introduction

As A new energy-efficient transportation style and a successful business model of collaborative consumption, car sharing has significantly enhanced our city's livability recently [1], [2]. In essence, car sharing provides a smart automobile rental service in which a registered customer (a.k.a user) can reserve and access (i.e., check in and check out using the mobile phone) shared vehicles for short-term or long-term use, without human intervention. Currently, car sharing services can be roughly categorized into two types: station-based car sharing or free-floating car sharing [3]. Station-based car sharing systems require customers to pick up and return vehicles at settled stations, while free-floating car sharing systems support peer-to-peer car sharing between any two customers without a fixed pick-up/drop-off position.

Compared with traditional car rental services, car sharing services obviously bring extra advantages. As they are always charged per time or per mile, a customer can make a flexible schedule regarding where and when she would like to pick up and return a shared vehicle [4]. They also benefit the environment by mitigating pollution and traffic congestion, since car sharing services advance the development of green-energy electric vehicles and reduce the number of private vehicles on the road [5], [6]. As a result, more companies have deployed shared cars, built car sharing services, and developed various mobile-based car sharing applications in different platforms, such as Enterprise CarShare[1], Car2go[2], and Zipcar[3]. Under this ecosystem, customers can conveniently download and install these Android/IOS applications from application stores, and utilize these applications to rent shared cars by performing simple operations on their smartphones.

Most of these applications require customers to take an essential step before enjoying car sharing services: identity uploading and verification. Specifically, a customer is required to upload a photo of her driving license (front and back) as well as a selfie of the customer holding it, and a car sharing service provider can review the personal identification to confirm that the customer has the right and ability to drive as a valid driver. The step is commonly indispensable as car sharing service providers need to check the driving qualification of the customer and trace the customer in case some bad situations happen. For instance, if a customer refuses to return a shared vehicle on time or leaves the shared vehicle in an unacceptable condition after one use, he will be assessed a certain fee. However, from a security and privacy

Manuscript received August 23, 2019; revised December 9, 2019; accepted February 17, 2020. Date of publication March 13, 2020; date of current version May 14, 2020. The review of this article was coordinated by Dr. P. Lin. *(Corresponding author: Cheng Huang.)*

Cheng Huang and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: c225huan@uwaterloo.ca; sshen@uwaterloo.ca).

Rongxing Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB E3B 5A3, Canada (e-mail: rlu1@unb.ca).

Jianbing Ni is with the Department of Electrical and Computing Engineering, Queen's University, Kingston, ON K7L 3N6, Canada (e-mail: jianbing.ni@queensu.ca).

Digital Object Identifier 10.1109/TVT.2020.2980777

[1][Online]. Available: https://www.enterprisecarshare.ca/

[2][Online]. Available: https://www.car2go.com/

[3][Online]. Available: https://www.zipcar.ca/

standpoint [7]–[9], this kind of necessary identity disclosure may lead to serious privacy concerns of customers. In reality, when a car sharing service provider is honest-but-curious as an internal adversary or has been compromised by an external adversary, a customer's privacy can be easily violated by analyzing the collected sensitive information [10]–[13]. The sensitive information includes real-time trajectories (through GPS on shared vehicles), pick-up and drop-off places, time duration of driving, etc. Since an adversary knows the real identity of the customer, he can link the information to a real person (corresponding to the customer) in the real world to further reconstruct her mobility patterns. Furthermore, modern laws (e.g., the EU General Data Protection Regulation (GDPR)) obligate service providers to better protect customers' privacy in the real-name system, by offering built-in privacy-preserving mechanisms [14]. Therefore, how to resolve the conflict between privacy and accountability becomes challenging in the car sharing scenario.

As the study of car sharing is still in its infancy, there are not many secure and privacy-enhancing schemes designed for this service [15]–[18]. The most related work is SePCAR [15], which proposes two basic approaches: one is designed based on a single trusted third-party authority (TTPA) and the other is designed from the secure multi-party computation (SMPC) [19]. Apparently, TTPA can protect customers' privacy and achieve accountability at the same time. Customers' private information can be stored at TTPA and be revealed as needed, nevertheless it still suffers from vulnerabilities like the single point of failure (i.e., the single TTPA is down accidentally or is compromised by an adversary). To tackle the issue, SePCAR also presents an SMPC-based approach where multiple fixed parties replace TTPA to manage the private information of customers and offer the accountability. Compared with the TTPA-based approach, the SMPC-based approach requires more time-consuming computations among multi-parties, and it is designed based on a non-collusion assumption where these parties cannot collude with each other.

Different from the existing works, we propose a decentralized, accountable, and privacy-preserving architecture for car sharing services in this paper, named DAPA. In DAPA, to avoid the single point of failure and build decentralized trust for customers, multiple validation servers are employed to replace a single TTPA. Each validation server is managed by an independent authority, and multiple authorities are organized as a group. The group is dynamic instead of fixed, i.e. after a time period, the group of authorities will be substituted by another group of authorities to improve the security level of the system due to the timeliness of the compromise attack. The motivation behind DAPA is to improve the fault tolerance of the car sharing service. Compared with other services, privacy protection and accountability are more necessary for the car sharing service. Multiple distributed authorities who manage customers' identities are substituted periodically such that attackers have more difficulties in compromising customers' privacy and break the accountability. To protect customers' privacy and achieve accountability simultaneously under the decentralized architecture, a new privacy-preserving identity management (PPIM) scheme is introduced as a basic module for DAPA. Through PPIM, customers' identities can be efficiently and secretly managed in a distributed and dynamic manner. As long as a majority of validation servers are honest during a time period, customers' identities are always hidden from car sharing service providers. With the help of validation servers, car sharing service providers can verify the validity of customers' hidden identities without revealing them and trace real identities of misbehaving customers. Specifically, there are three major technical challenges in designing PPIM.

Technical Challenges: First, considering that a customer's identity needs to be hidden, a trivial approach is to encrypt the identity before uploading. However, once the uploaded identity is encrypted, it would be difficult for a validation server to verify the validity of the customer's identity with the ciphertext. To enable identity validation, the following three properties should be guaranteed: i) (recoverable property) the ciphertext can be decrypted by the validation server; ii) (identity property) the plaintext is one registered customer's identity credential; iii) (legitimate property) the customer is a valid customer and has not been revoked. Second, since more than one validation server exist, the above-mentioned three properties should be verified in a distributed manner, i.e., the ability of verifying and recovering the customer's identity should be shared among multiple validation servers, which is challenging. Moreover, this ability should be verified as well, i.e., these validation servers can verify and ensure that they have the ability to accomplish identity verification and identity recovery. Third, validation servers are dynamic, which leads to the transferring of the ability of recovering a customer's identity from one set of validation servers to another set of validation servers after a time period, which is not straightforward. This process should also be verified by validation servers to detect malicious validation servers and ensure the correctness of transferring.

Contributions: The contributions of this paper are summarized as two-fold.

- A privacy-preserving identity management (PPIM) scheme is proposed. Through PPIM, a customer can outsource her encrypted identity to multiple dynamic validation servers for the purpose of decentralized identity management. These validation servers can verify the validity of the customer's encrypted identity without decrypting it based on a well-designed zero-knowledge proof protocol, recover the customer's real identity, and dynamically transferring the ability of identity recovery based on adaptive verifiable secret sharing/redistribution techniques. Although PPIM is a basic module of DAPA, it can also be integrated into other applications related to the identity management.
- A decentralized, accountable, and privacy-preserving architecture for car sharing services (DAPA) is proposed. DAPA is designed based on PPIM and other cryptographic primitives to preserve customers' privacy during the car sharing process while providing the accountability, i.e., DAPA enables a car sharing service provider to check a customer's driving qualification before the customer rents the car and to effectively trace the customer without a single trusted authority once the customer misbehaves. Detailed security analysis shows that DAPA can minimize privacy breaches as well as guarantees accountability. In addition,

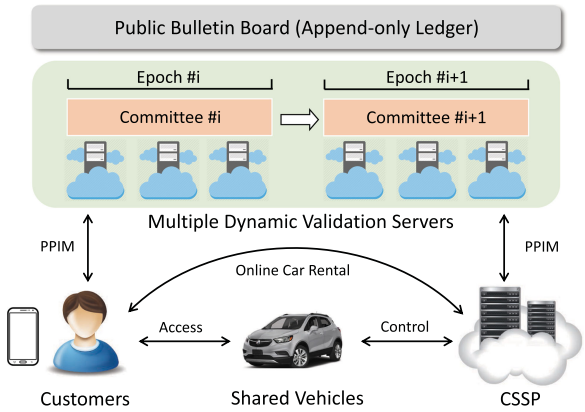


Fig. 1. System model.

performance evaluations via extensive simulations demonstrate that DAPA is efficient in terms of computational costs and communication overheads.

The remainder of this paper is organized as follows. In Section II, we first introduce the system model, the threat model, and the design goals. Next, we propose a privacy-preserving identity management scheme and present a decentralized, accountable, and privacy-preserving car sharing architecture in Section III. Subsequently, security analysis and performance evaluation are shown in Section IV and Section V, respectively. Finally, Section VI reviews the related work and Section VII draws the conclusion.

II. Models and Design Goals

In this section, we formalize the system model, the threat model, and the design goals.

A. System Model

In the system model, there exist five entities in a car sharing service, as shown in Fig. 1: a large number of customers, some shared vehicles, a car sharing service provider, multiple dynamic validation servers, and a public bulletin board.

- Customers: customers who install the car sharing application published by the car sharing service provider, can rent the nearby unoccupied vehicles at the car sharing station via simple operations using a smartphone. To ensure that a customer is an authorized driver, the customer needs to pass the identity verification at the service provider side. After being verified, the customer can reserve the vehicle and access the vehicle through the car sharing mobile application.
- Shared Vehicles: Vehicles are dispersed into the city and are under the management of the car sharing service provider. These vehicles can receive the control commands from the car sharing service provider remotely and then update its access privilege for different customers, i.e., a vehicle allows one and only one customer's access when this customer has successfully completed the car rental through the car sharing mobile application.
- Car Sharing Service Provider (CSSP): CSSP is a company, e.g., Zipcar, who possesses an online server to provide the car sharing service and publishes the corresponding mobile applications. It is also responsible for verifying the driving qualification of customers, deploying the vehicles in the city and managing these vehicles. Finally, the company can make a profit by charging customers based on mileage or time.
- Validation Servers (VSs): VSs can be regarded as dynamic and distributed servers. VSs are organized to form a fixed-size committee in a fixed time interval (a.k.a an epoch), and one specific VS is a *committee member* during this epoch. The committee members are dynamically replaced by another committee members after an epoch and they are responsible for identity management, i.e., verifying customers' driving qualification, showing the driving qualification to the car sharing service provider, and managing customers' real identities in a privacy-preserving way. When disputes arise between customers and the car sharing service provider, they can collaborate to recover the real identities of customers such that accountability is clear.
- Public Bulletin Board: An append-only ledger exists in the model, e.g., a public blockchain [8], where other entities can read/write the data. It can be regarded as a public bulletin board [20].

Communication Model: There exist private channels among customers and VSs such that customers can privately transmit data to VSs and VSs can share data with each other privately. The private channels can be straightforwardly implemented based on the mature Secure Sockets Layer protocol (SSL). Therefore, we omit the detailed construction of this part.

B. Threat Model

There exist two attacks from internal/external adversaries who make profits by selling personal information. First, VSs themselves are internal adversaries and can collude with each other to disclose a customer's identity, but these VSs cannot occupy a majority of committee members during an epoch. Second, an external adversary can compromise some honest VSs, but it cannot forecast VSs' change over time, corrupt a set of VSs in advance and therefore control a majority of the committee members during an epoch.

All in all, a majority of the committee members are honest during an epoch while some VSs can be compromised by internal or external adversaries and behave maliciously. In practice, these VSs can be different servers managed by different operators to limit the risk of most of the VSs being compromised and colluding against customers.

CSSP is assumed to be honest-but-curious in the system, that is, it may honestly provide the car sharing service for customers but may be curious to collect the personal information of customers and perform a deep analysis on customers' data which may reflect the customer's privacy. In other words, we assume that CSSP does not provide customers with malicious smartphone applications or malicious vehicles to monitor the customer since such attacks can be detected by third parties and cause the risk of reputation loss. Furthermore, CSSP can also collude with the malicious VSs.

From another point of view, customers cannot be fully trusted either, since some of them may misbehave during the car rental process, e.g., misbehaving customers may not return the vehicle or damage the shared vehicle unintentionally or intentionally after one use.

C. Design Goals

There exists a huge conflict between a customer's privacy requirement and CSSP's demand of accountability. On one hand, customers would like to prevent privacy leakage during the car sharing service. On the other hand, CSSP needs to have the ability to know the real identity of a customer so that it can review the customer's qualification of driving and claim the responsibility if the customer misbehaves. Hence, the following two security objectives should be satisfied simultaneously.

- *Customer Privacy:* The privacy of customers should be protected, which implies the anonymity and unlinkability of customers. More concretely, when a customer uses the car sharing application to rent a shared car online, her identity cannot be distinguished among all registered customers. When a customer uses the car sharing application to rent more than one shared cars online, her two renting records cannot be linked.
- *Accountability:* Customers should be held accountable for their behavior in the car sharing service, i.e., CSSP is able to check customers' driving qualification, recover the misbehaving customer's real identity, and revoke the misbehaving customer if necessary.

In addition to customer privacy and accountability, **usability** is also significant. The convenience and usability properties offered by the current car sharing service should be preserved. For instance, CSSP can easily verify the driving qualification of customers and customers can perform simple and straightforward operations to achieve online car rental.

In summary, our goal is to design an accountable and privacy-preserving car sharing architecture that offers strong privacy guarantees to customers as well as provides requisite accountability for CSSP.

III. Proposed DAPA

In this section, we propose a decentralized, accountable, and privacy-preserving car sharing architecture, named DAPA. We begin with two cryptographic primitives: the bilinear pairing technique and the zero-knowledge proof technique, and then present an overview of DAPA. Afterwards, we propose a privacy-preserving identity management (PPIM) scheme, serving as a basic module for DAPA. Finally, we show the detailed construction of DAPA.

A. Cryptographic Primitives

1) Bilinear Pairing: Let G_1, G_2 and G_T be three cyclic groups of prime order q with the multiplication. Let g_1, g_2 be generators of G_1 and G_2. Let $e : G_1 \times G_2 \rightarrow G_T$ be an asymmetric bilinear map and e has the following properties: i) Bilinearity: for all $(u, v) \in G_1 \times G_2$ and $(a, b) \in Z_q$, we have the relation $e(u^a, v^b) = e(u, v)^{ab}$; ii) Non-degeneracy: $e(g_1, g_2) \neq 1$; iii) Computability: There is an efficient algorithm to compute bilinear map $e(u^a, v^b) = e(u, v)^{ab}$; and iv) There is no efficiently computable homomorphism in either direction between G_1 and G_2. For more details, we refer the reader to [21].

2) Zero-Knowledge Proof: The zero-knowledge proof of knowledge allows the prover to generate a cryptographic proof with a corresponding statement, and the verifier can verify the proof to check the correctness of the statement. For easy understanding, we denote the zero-knowledge proof of knowledge ($ZkPoK$), similar to [22], where a prover convinces a verifier of knowledge of values $(a_1, \ldots, a_n)$ that satisfy a predicate $\mathbb{P}$ by $ZkPoK\{(a_1, \ldots, a_n)|\mathbb{P}(a_1, \ldots, a_n)\}$. For instance, $ZkPoK\{(a, B) : A = g_1^a \wedge C = e(A, B)\}$ denotes "zero-knowledge proof of knowledge of a and B such that $A = g_1^a$ and $C = e(A, B)$ holds". The Fiat-Shamir heuristic can be applied to turn the interactive zero-knowledge proof of knowledge into the non-interactive one in the random oracle model. For more details, we refer the reader to [23].

B. DAPA Overview

DAPA consists of five major phases: system setup, customer registration, car rental, car audit, and customer revocation.

- *System Setup:* The cryptographic parameters, key pairs, and public information for car sharing service are generated by CSSP.
- *Customer Registration:* A customer makes the registration at CSSP by providing the username, password, and relative driving license info. If the registration is successful, CSSP sends the identity credential back to the customer.
- *Car Rental:* Using a valid identity credential, a customer achieves the anonymous car rental via communicating with VSs (current committee members) and CSSP.
- *Car Audit:* With the help of VSs (current committee members), CSSP traces and reveals the real identity of a customer who misbehaves.
- *Customer Revocation:* CSSP revokes the misbehaving customer and does not accept these customers' car rental requests in the future.

Note that, the anonymous payments (car rental fee and car insurance fee) are not included in DAPA. If needed, the existing anonymous payment schemes like ZCASH [24] would be much helpful. Alternatively, the car sharing service can be an optional membership service. The valid customer who pays the membership fee during registration can enjoy the unlimited car sharing service without a rental fee or insurance fee.

C. Proposed PPIM

We first present PPIM, which is a key module that provides drivers' identity management for DAPA. PPIM allows VSs to manage real identities of any customers in a distributed and dynamic manner. It consists of six steps, namely, Parameter Generation (*PGen*), Identity Registration (*IDRegister*), Identity Hiding (*IDHide*), Identity Transferring (*IDTransfer*), Identity Recovery (*IDRecover*), and Identity Revocation (*IDRevoke*). In *PGen*, all the public parameters are generated and shared with the entities. In *IDRegister*, a customer registers herself at CSSP and obtains a valid identity credential. In *IDHide*, the

TABLE I
NOTATIONS FREQUENTLY USED IN PPIM

Notation	Definition
$\tilde{G}$, $\mathfrak{G}$, G_T	three groups that support bilinear maps
$l, \tilde{l}$	security parameters
p, p'	two primes that satisfy $p = 2p' + 1$
q, q'	two primes that satisfy $q = 2q' + 1$
n	a composite number $n = pq$
$\tilde{p}, \tilde{q}$	two primes satisfy $\tilde{p} = 2\tilde{q} + 1$
$H(), H'()$	two cryptographic hash functions
$\tilde{g}, \tilde{h}$	two generators of $\tilde{G}$
$\mathfrak{g}$	a generator of $\mathfrak{G}$
$e(.,.)$	a non-degradable bilinear mapping
$\bar{G}$	a cyclic subgroup $\bar{G} \subset Z^*_{\tilde{p}}$
τ	auxiliary information
(y, Y)	CSSP's private/public key
$(Cred, \sigma)$	the identity credential
$(y_1, y_2, y_3, Y_1, Y_2, Y_3)$	a customer's private/public key
(t, N)	threshold for identity management
S_{inv}	invalid customer list
(a, d)	a valid customer's witness
(u, w, v)	ciphertext of σ

customer uploads her identity credential to multiple VSs in a privacy-preserving manner and these VSs are organized as a group to manage the identity credential. A single VS cannot recover the identity credential but can verify the validity of the credential based on the zero-knowledge proof technique. In *IDTransfer*, the current group of validation servers transfers the identity management permission to another group of validation servers after a time period. In *IDRecover*, a majority of VSs in the group can cooperate with each other to recover the identity credential of a customer if necessary. In *IDRevoke*, a customer can be revoked by CSSP via revoking her identity credential and the credential becomes invalid for renting a shared car in the future. For easy understanding, Table I shows the notations frequently used in PPIM.

• *Parameter Generation (PGen):* This part is run by CSSP during system setup. CSSP can generate parameters as follows: i) p, p', q, q' are four prime numbers which satisfy $p = 2p' + 1$ and $q = 2q' + 1$. $l = |p| = |q|$ is the security parameter and $n = pq$; ii) g' is a random element in $Z^*_{n^2}$ and $g = (g')^{2n}$; iv) $\tilde{p}$ is a $\tilde{l}$-bit ($\tilde{l} \leq \frac{l}{2} - 1$) prime number that satisfies $\tilde{p} = 2\tilde{q} + 1$ and $\tilde{q}$ is also a prime number; v) $\tilde{G}$, $\mathfrak{G}$ and G_T are three bilinear groups of prime order $\tilde{p}$ and an asymmetric bilinear map $e : \tilde{G} \times \mathfrak{G} \rightarrow G_T$ exists; vi) $(\tilde{g}, \tilde{h})$ are two generators of $\tilde{G}$ and $\mathfrak{g}$ is a generator of $\mathfrak{G}$; vii) τ is a random number picked from $Z^*_{\tilde{p}}$ and $\bar{G} \subset Z^*_{\tilde{p}}$ is a cyclic group of prime order $\tilde{q}$; viii) H, H' are two cryptographic hash functions: $H : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^l$ and $H' : \{0,1\}^* \rightarrow \{0,1\}^{\tilde{l}}$; ix) its private/public key pair is $(y, Y = \mathfrak{g}^y)$, where $y \in_R Z^*_{\tilde{p}}$.

• *Identity Registration (IDRegister):* This part is run between CSSP and a customer during customer registration. If the registration is successful, the customer can obtain the identity credential from CSSP. Specifically, the customer sends the registration request to CSSP, and CSSP verifies the received information. As long as it is correct and legitimate, CSSP generates the identity credential for the customer as $Cred = \tilde{g}^{\frac{1}{y+\sigma}}$, where σ is chosen from $\bar{G}/\{-\tau\}$ and is unique for each customer. Then, $(\sigma, Cred)$ is sent back to the customer as the credential. The customer verifies the credential as $e(Cred, Y\mathfrak{g}^{\sigma}) \stackrel{?}{=} e(\tilde{g}, \mathfrak{g})$.

• *Identity Hiding (IDHide):* This part is run between a customer and VSs (current committee members) during car rental. The customer can upload an identifier, a ciphertext and a corresponding proof to the bulletin board. The proof indicates that the ciphertext possesses three properties:

– (Recoverable Property) The ciphertext can be decrypted to obtain a plaintext using a given private key.
– (Identity Property) The plaintext is one registered customer's identity credential.
– (Legitimate Property) The registered customer is a valid customer and has not been revoked.

Supposing that the current committee includes N VSs, the customer and VSs can perform the following steps.

– The customer randomly chooses the private keys as $(y_1, y_2, y_3) \in_R Z^*_{\tilde{p}}$ and generates the public keys as $(Y_1 = g^{y_1}, Y_2 = g^{y_2}, Y_3 = g^{y_3})$, and writes the public keys (Y_1, Y_2, Y_3) into the bulletin board.
– The customer distributes the private keys (y_1, y_2, y_3) to N VSs (current committee members) $\mathfrak{P} = (P_1, P_2, \ldots, P_N)$ with access structure (t, N), where $t = \lfloor \frac{N}{2} \rfloor + 1$ ($\lfloor value \rfloor$ means $value$ is rounded down). To distribute the private key y_1 (and y_2 and y_3), the customer and VSs follow the below stages synchronously in a sequential order and each stage has fixed duration.
  ∗ (Sharing Stage) In this stage, the customer chooses a random polynomial $f(x) = a_0 + a_1 x + \cdots + a_{t-1}x^{t-1}$ over $Z^*_{\tilde{p}}$ of degree $t - 1$. The customer sets $f(0) = a_0 = y_1$. The customer then computes the secret shadow $s_i = f(i)$ from $i = 1$ to N, and distributes s_i to every member $P_i \in \mathfrak{P}$ via the private channel. After that, the customer writes $\{g^{s_i}\}_{i=1}^{N}$ and $\{D_k = g^{a_k}\}_{k=1}^{t-1}$ into the bulletin board.
  ∗ (Complaining Stage) P_i verifies the shares it received from the customer. P_i checks if
$$g^{s_i} = Y_1 \cdot \prod_{k=1}^{t-1} (D_k)^{i^k}. \quad (1)$$
  If the check fails, P_i writes the complaint into the bulletin board in this stage.
  ∗ (Responding Stage) The customer, after checking the complaint from P_i, writes s_i that satisfies the Eq. (1) as a response into the bulletin board in this stage.
  ∗ (Confirming Stage) The customer is marked as disqualified if either more than t complaints are received or the response of a complaint falsifies the Eq. (1). Otherwise, P_i stores the secret shadow as s_i in this stage and confirms the sharing in the bulletin board.
– The customer downloads the latest accumulated value c (the value is defined in *IDRevoke*) and the revocation list including all invalid customers' credentials $S_{inv} = \{\sigma_1, \sigma_2, \ldots, \sigma_{n'}\}$. The customer calculates the witness (a, d) as $d = \prod_{k=1}^{n'} (\sigma_k + \tau) \bmod (\sigma + \tau)$ and $a = [\tilde{g}^{\prod_{k=1}^{n'} (\sigma_k + \tau) - d}]^{\frac{1}{\sigma + \tau}}$.

– The customer chooses a random number $r \in_R [0, n/4]$ and encrypts her identity credential σ as follows:

$$\begin{aligned} u &= g^r, \\ w &= Y_1^r (1+n)^\sigma, \\ v &= abs((Y_2 Y_3^{H(w,u)})^r). \end{aligned} \tag{2}$$

Note that, this is a ciphertext of an encryption that provides adaptive chosen ciphertext security [25].

– The customer generates a non-interactive zero-knowledge proof π, which proves three properties: 1) (u, w, v) is a valid ciphertext that is encrypted under the public keys (Y_1, Y_2, Y_3); 2) the ciphertext is an encryption of σ which is used for recovering the identity credential of the customer; 3) σ, included in the commitment, is a valid identity credential and has not been revoked by CSSP. The proof can be written as follows. γ and ϵ are two random values picked from $Z_{\tilde{p}}^*$.

$$\begin{aligned} &ZkPoK\{(r, \epsilon, \sigma, \gamma, a, d) : u^2 = g^{2r} \\ &\wedge w^2 = Y_1^{2r}(1+n)^{2\sigma} \wedge v^2 = (Y_2 Y_3^{H(w,u)})^{2r} \\ &\wedge A = \tilde{g}^\sigma \tilde{h}^\epsilon \wedge C = g^{\frac{\gamma}{y+\sigma}} \\ &\wedge e(a, \mathfrak{g}^\sigma \mathfrak{g}^\tau) e(\tilde{g}, \mathfrak{g})^d = e(c, \mathfrak{g}) \wedge d \neq 0\}. \end{aligned} \tag{3}$$

– The customer generates a random public key as the identifier and stores the corresponding private key. Then, the customer uploads the identifier ID_{user}, the ciphertext (u, w, v) and the corresponding proof π into the bulletin board as an identity record, such that her identity is hidden but publicly verified.

– VS (each current committee member) verifies the proof π and updates the state (approval or reject) of this identity record. If more than half committee members updates with success, the identity is successfully hidden. Otherwise, it fails.

• *Identity Transferring (IDTransfer):* This part is run between the current committee and the next committee at the end of an epoch. Since committee members change dynamically, the current committee should transfer the secrets (namely, the private keys (y_1, y_2, y_3)) maintained by themselves to the committee in the next epoch. N VSs follow the below stages synchronously in a sequential order to redistribute χ secrets (χ is the number of secret identities maintained by the current committee) to another $\hat{N}$ VSs with a new access structure $(\hat{t}, \hat{N})$, where $\hat{t} = \lfloor \frac{\hat{N}}{2} \rfloor + 1$.

– (Sharing Stage) Each VS in the current committee P_i chooses a random polynomial $\hat{f}_i(x) = \hat{a}_{i,0} + \hat{a}_{i,1} x + \cdots + \hat{a}_{i,\hat{t}-1} x^{\hat{t}-1}$ over $Z_{\tilde{p}}^*$ of degree $\hat{t}-1$. P_i sets $\hat{f}_i(0) = \hat{a}_{i,0} = s_i$ and writes $\hat{C}_{i,k} = g^{\hat{a}_{i,k}}$ from $k = 0$ to $\hat{t}-1$ into the bulletin board. P_i computes the shares $\hat{s}_{i,j} = \hat{f}_i(j)$ from $j = 1$ to $\hat{N}$ and sends $\hat{s}_{i,j}$ to each member in the next committee P_j via the private channel.

– (Complaining Stage) P_j verifies the shares it received from P_i. P_j checks if

$$g^{\hat{s}_{i,j}} = \prod_{k=0}^{\hat{t}-1} (\hat{C}_{i,k})^{j^k}. \tag{4}$$

If the check fails, P_j writes the complaint against P_i into the bulletin board in this stage.

– (Responding Stage) P_i, after checking the complaint from P_j, writes $\hat{s}_{i,j}$ that satisfies the Eq. (4) as a response, into the bulletin board in this stage.

– (Qualifying Stage) ① P_i is marked as disqualified if either more than $\hat{t}$ complaints are received or the response of a complaint falsifies the Eq. (4); ② P_j tests whether $\hat{C}_{i,0}$ is equal to g^{s_i}. If not, P_i is marked as disqualified; ③ P_j builds the same set of non-disqualified members $QUAL$. If the size of $QUAL$ is larger than t, P_j chooses the first t members in $QUAL$, as the set $QUAL_t$.

– (Transferring Stage) P_j calculates its new secret shadow as $s_j = \sum_{i \in QUAL_t} b_i \hat{s}_{i,j}$ where $b_i = \prod_{x \in QUAL_t, x \neq i} \frac{x}{x-i}$, and writes g^{s_j} into the bulletin board.

– (Deleting Stage) P_i deletes its old secret shadow s_i.

• *Identity Recovery (IDRecover):* This part is run between VSs and CSSP during car audit. The committee members can recover the private keys used for encrypting identity credential by contributing their secret shadows. Specifically, if an identity of a customer needs to be recovered, each committee member P_i (or P_j) first encrypts its secret shadow s_i (or s_j) using CSSP's public key (based on any public-key cryptosystem, e.g., ElGamal encryption), writes the ciphertext into the bulletin board, and informs CSSP that the recovery operation is completed. Then, only CSSP can decrypt the ciphertext to obtain the secret shadow and verify the correctness of the secret shadow by checking g^{s_i} (or g^{s_j}). After receiving t (or $\hat{t}$) secret shadows, the secret y_1 (and y_2 and y_3) can be recovered by CSSP as follows.

$$y_1 = \sum_{i=1}^{t} \left(s_i \cdot \prod_{k=1, k \neq i}^{t} \frac{k}{k-i} \right). \tag{5}$$

To decrypt the identity of the customer, CSSP checks $u^{2(y_2 + H(w,u) y_3)} = v^2$ and computes $m' = (w / u^{y_1}) \bmod n^2$. Finally, the plaintext is $\sigma = (m'-1)/n$, which is the identity credential.

• *Identity Revocation (IDRevoke):* This part is run by CSSP during customer revocation. CSSP can revoke a customer by adding the invalid customer's identity credential σ into a revocation list S_{inv} and updating an accumulator value c. Particularly, CSSP creates an empty accumulator c as $c = \tilde{g}$ in the beginning. When a customer is invalid, CSSP adds his identity credential into the accumulator as $c = c^{\sigma+\tau}$ and also adds his credential σ into the revocation list S_{inv}. Finally, CSSP publishes the latest accumulator c and revocation list S_{inv} to the bulletin board.

D. Detailed Construction of DAPA

• *System Setup:* During this phase, CSSP runs *PPIM.PGen* to generate the public cryptographic parameters $Params$ and the

private key $y \in Z_{\tilde{p}}^*$.

$$Params = \{l, \tilde{l}, n, g, \tilde{p}, \tilde{q}, \tilde{G}, \mathfrak{G}, G_T, e, \tilde{g}, \tilde{h}, \mathfrak{g}, \bar{G}, H, H,' \tau, Y\}.$$

Eventually, CSSP publishes $Params$ and stores y in its local storage.

• *Customer Registration:* A customer registers herself at CSSP. Following *PPIM.IDRegister*, the customer sends a unique username/password (note that, they are used for re-generating/retrieving the credential if the credential is missing accidentally) and the corresponding driving qualification info, e.g., the photocopy of a valid driver license, to CSSP. After receiving the registration request, CSSP verifies the validity of the driving qualification info. If the information is correct and legitimate, CSSP stores the real identity of this customer and generates a unique identity credential $(\sigma, Cred)$. CSSP then returns the credential $(\sigma, Cred)$ to the customer. Otherwise, CSSP returns with failure. After receiving the credential, the customer verifies the credential $(\sigma, Cred)$. If it passes the verification, the customer stores the credential $(\sigma, Cred)$. Otherwise, the customer's registration fails.

• *Car Rental:* A customer achieves online car rental and obtains a code to access the shared car through CSSP and VSs. Following *PPIM.IDHide*, the customer generates the key pairs $\{(y_1, Y_1), (y_2, Y_3), (y_2, Y_3)\}$ and publishes the public keys $\{Y_1, Y_2, Y_3\}$ to VSs. The customer then shares the private key $\{y_1, y_2, y_3\}$ with the distributed VSs and VSs store the received secret shadows. Subsequently, the customer generates the witness (a, d), encrypts her identity as (u, w, v), generates a proof of identity credential π, generates a unique public key Key_{user}, generates an identifier ID_{user} (note that, the identifier is another unique public key that generated by the customer), and writes $\{ID_{user}, Key_{user}, (u, w, v), \pi\}$ to the bulletin board as an identity record. The customer stores the corresponding private keys of Key_{user} and ID_{user}. Next, she sends a verification request to the current committee members. The current committee (i.e., each individual committee member) verifies the uploaded proof, and updates the state of the identity record (approval or reject) at the bulletin board as well as sends the response back to the customer. After confirming that the record's state is updated (approval), the customer sends a car rental request to CSSP, which includes the identifier ID_{user}, a signature (i.e., the customer uses the private key corresponding to the identifier to generate the signature based on any secure signature scheme, e.g., BonehLynnShacham signature [21]), the shared car's information, and the rental duration. After receiving the request, CSSP locates the identifier ID_{user} at the bulletin board, verifies the signature to ensure that the record belongs to the requester, and checks the state of this record. If the state is approval, CSSP updates the code for the shared car, encrypts the car access code using the public key Key_{user} based on any public-key cryptosystem, e.g., ElGamal encryption, writes the encrypted code into the bulletin board, creates a car sharing record, and responds to the customer. Otherwise, it rejects the request. The customer downloads the encrypted code from the bulletin board and decrypts it to obtain the code using the private key corresponding to the public key Key_{user}. Finally, the customer uses the code to unlock the shared car at the car sharing station.

Meanwhile, since the committee members change periodically (every epoch), following *PPIM.IDTransfer*, the previous committee transfers the secret shadows to the next committee at the end of each epoch. When a customer returns the car, the customer parks the shared car at any car sharing station and confirms the return operation by sending the return request to CSSP. If the car is returned properly, CSSP then updates the code for the shared car and informs the current committee about the accomplishment and the identity record related to the car rental transaction. The committee members delete the stored secret shadows as well as update the state of the identity record (accomplishment). After confirming that the record's state is updated, CSSP confirms the return by sending the return response to the customer. Otherwise, CSSP goes to the car audit phase.

• *Car Audit:* CSSP recovers the identity of a customer who rents a shared car but misbehaves. Concretely, CSSP uploads the evidence to the bulletin board, pointing to the customer's identity record based on the unique identifier ID_{user}, and updates the state of the record (evil). CSSP then sends an audit request to VSs. After receiving the audit request, the current committee members check the evidence stored in the bulletin board. If the evidence exists and is correct, following *PPIM.IDRecover*, the current committee members release the encrypted secret shadows, update the state of the record (release), and send the response back to CSSP. Afterwards, CSSP downloads the encrypted secret shadows and encrypted identity from the bulletin board, and recovers the real identity of the customer via decryption.

• *Customer Revocation:* CSSP revokes a customer. Following *PPIM.IDRevoke*, CSSP updates the customer revocation list S_{inv} and the accumulator c used for identity verification. When a customer is revoked, he cannot pass the identity verification during identity registration and identity hiding. That is, the customer is forbidden to use the car sharing service.

IV. Security Analysis

In this section, we first analyze the proposed ZkPoK, i.e., Eq. (3), to show its completeness, special soundness, and special honest verifier zero-knowledge, and then analyze how PPIM achieves privacy preservation and accountability. As DAPA's core component is PPIM, the security of DAPA can be naturally reduced to the security of PPIM.

A. Security Analysis of ZkPoK

A ZkPoK based on the Σ protocol has three properties [23]: 1) *(Completeness)* if a prover ($\mathcal{P}$) and a verifier ($\mathcal{V}$) follow the ZkPoK protocol on input a public input x and a private input w, where $(x, w) \in R$ and R is a non-deterministic polynomial time relation, $\mathcal{V}$ always accepts $\mathcal{P}$'s proof; 2) *(Special Soundness)* for any x and any pair of accepting conversations on input x with different random challenges ch and ch' ($ch \neq ch'$), an extractor can efficiently extract w such that $(x, w) \in R$; and

3) *(Special Honest Verifier Zero-Knowledge)* there is a polynomial time simulator $\mathcal{S}$, which on input x and a random challenge ch, outputting an accepting conversation, with the same probability distribution as conversations between the honest prover and the honest verifier on input x.

Lemma 1: Eq. (3) satisfies completeness, special soundness, and special honest verifier zero-knowledge.

Proof. (Completeness) To prove the completeness, the proof should be transformed into another (equivalent) proof. The reason that the transformation is needed is that the proof is designed based on Σ protocol which only supports zero-knowledge proof of discrete log. To achieve the transformation, specifically, the following auxiliaries should be generated at the beginning. The customer as the prover chooses three independent generators of $\tilde{G}$: $\tilde{g}_1$, $\tilde{g}_2$ and $\tilde{g}_3$, and computes $A = \tilde{g}^{\sigma}\tilde{h}^{\epsilon}$, $C = Cred^{\gamma}$, $\theta_1 = \sigma\beta_1$, $\theta_2 = \sigma\beta_2$, $\theta_3 = d\beta_3$, $\theta_4 = d\beta_4$, $B_1 = \tilde{g}^{\beta_1}\tilde{h}^{\beta_2}$, $B_2 = a\tilde{h}^{\beta_1}$, $B_3 = \tilde{g_1}^{\beta_3}\tilde{g}_2^{\beta_4}$, $B_4 = \tilde{g}_3^{\theta_3}$, where $\beta_1, \beta_2, \beta_3, \beta_4$ are chosen from $Z_{\tilde{p}}^*$. Finally, the new proof can be represented as follows.

$$
\begin{aligned}
&ZkPoK\{(r, \epsilon, \sigma, \gamma, d, \beta_1, \beta_2, \beta_3, \beta_4, \theta_1, \theta_2, \theta_3, \theta_4) :\\
&u^2 = g^{2r} \wedge w^2 = Y_1^{2r}(1+n)^{2\sigma} \wedge v^2 = (Y_2 Y_3^{H(u,w)})^{2r}\\
&\wedge \frac{e(B_2, \mathfrak{g})^{\tau}}{e(c, \mathfrak{g})} = e(\tilde{g}, \mathfrak{g})^{-d} e(\tilde{h}, \mathfrak{g})^{\theta_1} e(\tilde{h}, \mathfrak{g})^{\tau\beta_1} e(B_2, \mathfrak{g})^{-\sigma}\\
&\wedge A = \tilde{g}^{\sigma}\tilde{h}^{\epsilon} \wedge C = g^{\frac{\gamma}{y+\sigma}} \wedge B_1 = \tilde{g}^{\beta_1}\tilde{h}^{\beta_2} \wedge 1 = B_1^{-\sigma}\tilde{g}^{\theta_1}\tilde{h}^{\theta_2}\\
&\wedge B_3 = \tilde{g_1}^{\beta_3}\tilde{g}_2^{\beta_4} \wedge 1 = B_3^{-d}\tilde{g_1}^{\theta_3}\tilde{g}_2^{\theta_4} \wedge B_4 = \tilde{g}_3^{\theta_3} \wedge d \neq 0\}.
\end{aligned}
$$

The above protocol is a standard Σ protocol. Following the Σ protocol, the customer first chooses random $\dot{r}, \dot{\epsilon}, \dot{\sigma}, \dot{\gamma}, \dot{d}, \dot{\beta}_1, \dot{\beta}_2, \dot{\beta}_3, \dot{\beta}_4, \dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3, \dot{\theta}_4 \in Z_{\tilde{p}}^*$ and computes

$$
\begin{aligned}
&\dot{u} = g^{2\dot{r}}, \dot{w} = Y_1^{2\dot{r}}(1+n)^{2\dot{\sigma}}, \dot{v} = (Y_2 Y_3^{H(u,w)})^{2\dot{r}},\\
&\dot{A} = \tilde{g}^{\dot{\sigma}}\tilde{h}^{\dot{\epsilon}}, \dot{C} = e(C, \mathfrak{g})^{-\dot{\sigma}} e(\tilde{g}, \mathfrak{g})^{\dot{\gamma}},\\
&\dot{B_{1,1}} = \tilde{g}^{\dot{\beta}_1}\tilde{h}^{\dot{\beta}_2}, \dot{B_{1,2}} = B_1^{-\dot{\sigma}}\tilde{g}^{\dot{\theta}_1}\tilde{h}^{\dot{\theta}_2},\\
&\dot{B_{3,1}} = \tilde{g_1}^{\dot{\beta}_3}\tilde{g}_2^{\dot{\beta}_4}, \dot{B_{3,2}} = B_3^{-\dot{d}}\tilde{g_1}^{\dot{\theta}_3}\tilde{g}_2^{\dot{\theta}_4}, \dot{B}_4 = \tilde{g}_3^{\dot{\theta}_3},\\
&\dot{D} = e(\tilde{g}, \mathfrak{g})^{-\dot{d}} e(\tilde{h}, \mathfrak{g})^{\dot{\theta}_1} e(\tilde{h}, \mathfrak{g})^{\tau\dot{\beta}_1} e(B_2, \mathfrak{g})^{-\dot{\sigma}}.
\end{aligned}
$$

Next, the customer calculates the challenge $ch = H'(\tilde{g}||g||A||\dot{A})$ based on Fiat-Shamir heuristic [23]. Afterwards, the customer calculates the witnesses $\ddot{r} = \dot{r} - ch \cdot r$, $\ddot{\epsilon} = \dot{\epsilon} - ch \cdot \epsilon$, $\ddot{\sigma} = \dot{\sigma} - ch \cdot \sigma$, $\ddot{\gamma} = \dot{\gamma} - ch \cdot \gamma$, $\ddot{d} = \dot{d} - ch \cdot d$, $\ddot{\beta_1} = \dot{\beta}_1 - ch \cdot \beta_1$, $\ddot{\beta_2} = \dot{\beta}_2 - ch \cdot \beta_2$, $\ddot{\beta_3} = \dot{\beta}_3 - ch \cdot \beta_3$, $\ddot{\beta_4} = \dot{\beta}_4 - ch \cdot \beta_4$, $\ddot{\theta_1} = \dot{\theta}_1 - ch \cdot \theta_1$, $\ddot{\theta_2} = \dot{\theta}_2 - ch \cdot \theta_2$, $\ddot{\theta_3} = \dot{\theta}_3 - ch \cdot \theta_3$, $\ddot{\theta_4} = \dot{\theta}_4 - ch \cdot \theta_4$. The customer finally sends the proof as follows.

$$
\begin{aligned}
\pi = \Big\{&A, C, B_1, B_2, B_3, B_4, \dot{u}, \dot{w}, \dot{v}, \dot{A}, \dot{C}, \dot{B_{1,1}}, \dot{B_{1,2}}, \dot{B_{3,1}},\\
&\dot{B_{3,2}}, \dot{B}_4, \dot{D}, \ddot{r}, \ddot{\epsilon}, \ddot{\sigma}, \ddot{\gamma}, \ddot{d}, \ddot{\beta}_1, \ddot{\beta}_2, \ddot{\beta}_3, \ddot{\beta}_4, \ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3, \ddot{\theta}_4\Big\}.
\end{aligned}
$$

After receiving the proof, VS as the verifier computes $ch = H'(\tilde{g}||g||A||\dot{A})$ and verifies the proof by checking whether the following relations hold.

$$
\begin{aligned}
&\dot{u} = u^{2\cdot ch} g^{2\ddot{r}}, \dot{w} = w^{2\cdot ch} Y_1^{2\ddot{r}}(1+n)^{2\ddot{\sigma}},\\
&\dot{v} = v^{2\cdot ch}(Y_2 Y_3^{H(u,w)})^{2\ddot{r}},\\
&\dot{A} = A^{ch}\tilde{g}^{\ddot{\sigma}}\tilde{h}^{\ddot{\epsilon}}, \dot{C} = e(C, Y)^{ch} e(C, \mathfrak{g})^{-\dot{\sigma}} e(\tilde{g}, \mathfrak{g})^{\ddot{\gamma}}, B_4 \neq 1,\\
&\dot{B_{1,1}} = B_1^{ch}\tilde{g}^{\ddot{\beta}_1}\tilde{h}^{\ddot{\beta}_2}, \dot{B_{1,2}} = 1^{ch} B_1^{-\ddot{\sigma}}\tilde{g}^{\ddot{\theta}_1}\tilde{h}^{\ddot{\theta}_2}, v = abs(v),\\
&\dot{B_{3,1}} = B_3^{ch}\tilde{g_1}^{\ddot{\beta}_3}\tilde{g}_2^{\ddot{\beta}_4}, \dot{B_{3,2}} = 1^{ch} B_3^{-\ddot{d}}\tilde{g_1}^{\ddot{\theta}_3}\tilde{g}_2^{\ddot{\theta}_4}, \dot{B}_4 = B_4^{ch}\tilde{g}_3^{\ddot{\theta}_3},\\
&\dot{D} = \left(\frac{e(B_2, \mathfrak{g})^{\tau}}{e(c, \mathfrak{g})}\right)^{ch} e(\tilde{g}, \mathfrak{g})^{-\ddot{d}} e(\tilde{h}, \mathfrak{g})^{\ddot{\theta}_1} e(\tilde{h}, \mathfrak{g})^{\tau\ddot{\beta}_1} e(B_2, \mathfrak{g})^{-\ddot{\sigma}}.
\end{aligned}
$$

If any of them does not hold, the verification fails. Otherwise, the proof passes the verification. The completeness is guaranteed.

(Special Soundness): We assume that the extractor input consists of two transcripts, i.e.,

$$
\begin{aligned}
&\{ch, \ddot{r}, \ddot{\epsilon}, \ddot{\sigma}, \ddot{\gamma}, \ddot{d}, \ddot{\beta}_1, \ddot{\beta}_2, \ddot{\beta}_3, \ddot{\beta}_4, \ddot{\theta}_1, \ddot{\theta}_2, \ddot{\theta}_3, \ddot{\theta}_4\},\\
&\{ch,' \ddot{r},' \ddot{\epsilon},' \ddot{\sigma},' \ddot{\gamma},' \ddot{d},' \ddot{\beta}_1,' \ddot{\beta}_2,' \ddot{\beta}_3,' \ddot{\beta}_4,' \ddot{\theta}_1,' \ddot{\theta}_2,' \ddot{\theta}_3,' \ddot{\theta}_4'\}.
\end{aligned}
$$

The first transcript is the original transcript in the proof while the second transcript is different from the first one, e.g. $\ddot{r}' = \dot{r} - ch' \cdot r$ and $\ddot{\epsilon}' = \dot{\epsilon} - ch' \cdot \epsilon$. The only difference is that the challenges ch and ch' is different. If VS accepts both transcripts, the witnesses for the statement in Eq. (3) can be computed and extracted by the following equations.

$$
\begin{aligned}
&r = \frac{\ddot{r} - \ddot{r}'}{ch - ch'}, \epsilon = \frac{\ddot{\epsilon} - \ddot{\epsilon}'}{ch - ch'}, \sigma = \frac{\ddot{\sigma} - \ddot{\sigma}'}{ch - ch'}, \gamma = \frac{\ddot{\gamma} - \ddot{\gamma}'}{ch - ch'},\\
&d = \frac{\ddot{d} - \ddot{d}'}{ch - ch'}, \beta_1 = \frac{\ddot{\beta}_1 - \ddot{\beta}_1'}{ch - ch'}, \beta_2 = \frac{\ddot{\beta}_2 - \ddot{\beta}_2'}{ch - ch'}, \gamma = \frac{\ddot{\beta}_3 - \ddot{\beta}_3'}{ch - ch'},\\
&\beta_4 = \frac{\ddot{\beta}_4 - \ddot{\beta}_4'}{ch - ch'}, \theta_1 = \frac{\ddot{\theta}_1 - \ddot{\theta}_1'}{ch - ch'}, \theta_2 = \frac{\ddot{\theta}_2 - \ddot{\theta}_2'}{ch - ch'}, \theta = \frac{\ddot{\theta}_3 - \ddot{\theta}_3'}{ch - ch'},\\
&\theta = \frac{\ddot{\theta}_3 - \ddot{\theta}_3'}{ch - ch'}.
\end{aligned}
$$

(Special Honest Verifier Zero-knowledge): We construct a simulator $\mathcal{S}$ who is given a random challenge ch. It randomly chooses $\ddot{r},' \ddot{\epsilon},' \ddot{\sigma},' \ddot{\gamma},' \ddot{d},' \ddot{\beta}_1,' \ddot{\beta}_2,' \ddot{\beta}_3,' \ddot{\beta}_4,' \ddot{\theta}_1,' \ddot{\theta}_2,' \ddot{\theta}_3,' \ddot{\theta}_4' \in Z_{\tilde{p}}^*$ and generates the conversation similarly which is an accepting conversation. In other words, the simulator can utilize these random numbers to generate another conversation as follows. For example, $\dot{u}' = u^{2\cdot ch} g^{2\ddot{r}'}$ and $\dot{w} = w^{2\cdot ch} Y_1^{2\ddot{r}'}(1+n)^{2\ddot{\sigma}'}$. $\dot{u}'$ cannot be distinguished from $\dot{u}$, and $\dot{w}'$ cannot be distinguished from $\dot{w}$ due to randomness.

$$
\begin{aligned}
\Big\{&\dot{u},' \dot{w},' \dot{v},' \dot{A},' \dot{C},' \dot{B_{1,1}},' \dot{B_{1,2}},' \dot{B_{3,1}},' \dot{B_{3,2}},' \dot{B}_4,' \dot{D},' \ddot{r},'\\
&\ddot{\epsilon},' \ddot{\sigma},' \ddot{\gamma},' \ddot{d},' \ddot{\beta}_1,' \ddot{\beta}_2,' \ddot{\beta}_3,' \ddot{\beta}_4,' \ddot{\theta}_1,' \ddot{\theta}_2,' \dot{\theta}_3,' \ddot{\theta}_4'\Big\}.
\end{aligned}
$$

Obviously, it is indistinguishable from the conversation which is generated by the honest prover.

B. Security Analysis of PPIM

In this section, a security definition for PPIM is given based on a simulation-based model, in a similar sense to the model adopted by [26]. First we summarize the idea of the security analysis.

In the real world, all entities communicate via PPIM while in the idea world, all entities communicate via a trusted party $\mathcal{T}$,

who handles the outputs and the inputs of all entities and achieves the functionality provided by PPIM. There exists an adversary, $\mathcal{A}$, who controls the same entities (e.g., malicious customers and honest-but-curious CSSP) in the real world and the ideal world. Also, there exists a probabilistic polynomial-time (PPT) algorithm, the environment $\mathcal{E}$, that provides the inputs to all entities and schedules the interaction among entities. $\mathcal{A}$ can freely communicate with $\mathcal{E}$. We adopt a static model (during one epoch) and assume the number of entities and whether they are honest or not are fixed before the system starts. We utilize an event to denote the execution of a functionality, and there exist six events: INIT, REG, HIDE, TRANS, ROC, and REV, corresponding to six parts of PPIM. All communications with $\mathcal{T}$ are not anonymous, i.e., $\mathcal{T}$ knows the identity of the entity who communicates with it, while the communication between honest entities is not observed by $\mathcal{A}$, which can be achieved by the anonymous network, e.g., Tor network [27].

• INIT. The system begins when $\mathcal{E}$ specifies the number of honest/malicious customers and VSs in the system. CSSP is honest-but-curious in the system.

– *Real World.* CSSP generates its key pair (spk, ssk). The public key spk is published to all entities in the system.
– *Ideal World.* $\mathcal{T}$ initializes a database DB, which is used for storing the registration status and storing the identity of the customer.

• REG. $\mathcal{E}$ instructs a customer to register with CSSP.

– *Real World.* The customer sends a registration request to CSSP, and CSSP responds to the customer and stores the customer's registration status. If the customer has already obtained an identity credential, CSSP would reject the request. Since this procedure is not anonymous in the view of CSSP, CSSP can identify duplicated requests from the same customer.
– *Ideal World.* The customer sends a registration request to $\mathcal{T}$. $\mathcal{T}$ then informs CSSP a customer would like to register and whether the customer has registered before. CSSP responds to $\mathcal{T}$, and $\mathcal{T}$ forwards the response to the customer. If CSSP accepts the request, i.e., the customer has not registered before, $\mathcal{T}$ stores the registration status of the customer in DB.

• HIDE. $\mathcal{E}$ instructs a customer to hide her identity through communicating with VSs.

– *Real World.* The customer generates the secret for each VS and encrypts her identity credential. Then the customer uploads the encrypted identity and a proof to the public bulletin board, and sends an identity hiding request to VSs. VSs verify the proof and update the state at the public bulletin board.
– *Ideal World.* The customer sends an identity hiding request to $\mathcal{T}$. $\mathcal{T}$ verifies the identity, and sends a bit to each VS indicating whether the customer is valid registered customer and has not been revoked. Each VS replies with the state to $\mathcal{T}$, and $\mathcal{T}$ then updates the state at the public bulletin board. If the state is approval, $\mathcal{T}$ stores the real identity of the customer in DB.

• TRANS. $\mathcal{E}$ instructs VSs in the current committee to transfer the secret belonging to a customer to VSs in the next committee.

– *Real World.* Each VS in the current committee redistributes the secret belonging to a customer to each VS in the next committee. Each VS in the next committee verifies the secret afterwards.
– *Ideal World.* Each VS in the current committee sends the secret transferring request to $\mathcal{T}$. $\mathcal{T}$ updates the identity credential of the customer in DB, and sends a bit indicating whether the transferring is successful or not to each VS in the current committee and next committee.

• ROC. $\mathcal{E}$ instructs CSSP to recover the identity of a customer through communicating with VSs.

– *Real World.* CSSP sends the identity recovery request, corresponding to an HIDE event initiated by a customer such that a majority of VSs output success, to each VS. Each VS replies with its secret and CSSP utilizes the secret to decrypt the encrypted identity of the customer.
– *Ideal World.* CSSP sends the identity recovery request to $\mathcal{T}$. $\mathcal{T}$ locates the real identity of the customer in DB. $\mathcal{T}$ informs each VS CSSP would like to recover a customer's identity. Each VS responds to $\mathcal{T}$ with a bit indicating whether the recovery is approved or not. If a majority of VSs agree to recover the identity, $\mathcal{T}$ replies the customer's identity back to CSSP. Then, CSSP recovers the identity of the customer.

• REV. $\mathcal{E}$ instructs CSSP to revoke a customer.

– *Real World.* CSSP adds the identity credential of the customer into the accumulator c and the revocation list S_{inv}. CSSP updates the latest accumulator and revocation list at the public bulletin board.
– *Ideal World.* CSSP sends the latest accumulator and revocation list to $\mathcal{T}$, and $\mathcal{T}$ updates them at public bulletin board and delete the customer in its database.

In the ideal world, PPIM provides all the desired security properties. First, all the events, in the view of CSSP and VSs, are anonymous. $\mathcal{T}$ just informs VSs some anonymous customers would like to hide their identities and the real identities are only maintained by $\mathcal{T}$. Thus, customer privacy is guaranteed. Second, $\mathcal{T}$ verifies whether the customer is a valid registered customer and has not been revoked, and $\mathcal{T}$ can recover the real identity of the customer and revoke a customer by deleting the customer in its database such that accountability is assured. Real world PPIM is secure if its behavior is the same as the ideal world PPIM. Thus, assuming $negl(\lambda)$ is a negligible function in security parameter, the following definition of security can be given.

Definition 1: Let $\mathsf{Real}_{\mathcal{E},\mathcal{A}}(\lambda)$ (resp. $\mathsf{Ideal}_{\mathcal{E},\mathcal{S}}(\lambda)$) be the probability that $\mathcal{E}$ outputs 1 when run in the real world (resp. ideal world) with adversary $\mathcal{A}$ (resp. $\mathcal{S}$ having black-box access to $\mathcal{A}$). PPIM is secure if for all PPT algorithms $\mathcal{E}$, $\mathcal{A}$, the following expression holds:

$$|\mathsf{Real}_{\mathcal{E},\mathcal{A}}(\lambda) - \mathsf{Ideal}_{\mathcal{E},\mathcal{S}}(\lambda)| = negl(\lambda).$$

We analyze the security of PPIM based on the following lemmas handling the relevant combinations of entities controlled by the adversary. The analysis is divided into two cases according to the subset of entities controlled by $\mathcal{A}$. The first case is proven to achieve customer privacy and the second case is proven to achieve accountability.

Lemma 2 (Customer Privacy): For all PPT environments $\mathcal{E}$ and all real world adversaries $\mathcal{A}$ controlling CSSP, a subset of customers, and a subset of VSs (less than half VSs), there exists an ideal world simulator $\mathcal{S}$ which satisfies $|\mathsf{Real}_{\mathcal{E},\mathcal{A}}(\lambda) - \mathsf{Ideal}_{\mathcal{E},\mathcal{S}}(\lambda)| = negl(\lambda)$.

Proof Sketch. A simulator $\mathcal{S}$ is defined which interacts with $\mathcal{E}$ as an ideal world adversary, and meanwhile has black-box access to a real world adversary $\mathcal{A}$. Note that the output of $\mathcal{S}$ is always indistinguishable to the output of $\mathcal{A}$ as long as the following conditions are satisfied. During an HIDE event, $\mathcal{S}$ represents the dishonest VS to $\mathcal{T}$ and represents the honest customer/VS to $\mathcal{A}$. The simulation fails if $\mathcal{A}$ can recover the identity credential σ corresponding to the ciphertext (u, w, v). This happens with negligible probability under the Decision Composite Residuosity (DCR) assumption. The security proof is similar to the proof of the adaptive chosen ciphertext security property of the encryption [25]. The simulation also fails if $\mathcal{A}$ can distinguish two proofs π and π'. This happens with negligible probability due to the zero-knowledgeness of the ZkPoK, which has been proven in Lemma 1. In addition, the simulation fails if $\mathcal{A}$ can break the the confidentiality of the t out of N secret sharing [28]. This happens with negligible probability under the Discrete Log (DL) assumption. Intuitively, since the share shadow obtained by each VS is distributed uniformly at random, so it does not contain any information about the secret. During a TRANS event, $\mathcal{S}$ represents the dishonest VS to $\mathcal{T}$ and represents the honest VS to $\mathcal{A}$. The simulation fails if $\mathcal{A}$ can break the the confidentiality of the secret redistribution [29]. This happens with negligible probability under the DL assumption.

Lemma 3 (Accountability): For all PPT environments $\mathcal{E}$ and all real world adversaries $\mathcal{A}$ controlling a subset of customers and a subset of VSs (less than half VSs), there exists an ideal world simulator $\mathcal{S}$ which satisfies $|\mathsf{Real}_{\mathcal{E},\mathcal{A}}(\lambda) - \mathsf{Ideal}_{\mathcal{E},\mathcal{S}}(\lambda)| = negl(\lambda)$.

Proof Sketch. A simulator $\mathcal{S}$ is defined which interacts with $\mathcal{E}$ as an ideal world adversary, and meanwhile has black-box access to a real world adversary $\mathcal{A}$. Note that the output of $\mathcal{S}$ is always indistinguishable to the output of $\mathcal{A}$ as long as the following conditions are satisfied. During a REG event, $\mathcal{S}$ represents the dishonest customer to $\mathcal{T}$ and represents the honest CSSP to $\mathcal{A}$. The simulation fails if $\mathcal{A}$ can forge a valid identity credential $(\sigma, Cred = \tilde{g}^{\frac{1}{y+\sigma}})$. This happens with negligible probability under the q-Strong Diffie-Hellman (q-SDH) assumption. The security proof is similar to the proof of the existential unforgeability property of the Boneh-Boyen short signature [30]. During an HIDE event, $\mathcal{S}$ represents the dishonest customer to $\mathcal{T}$ and represents the honest CSSP to $\mathcal{A}$. The simulation fails if $\mathcal{S}$ fails to extract from $\mathcal{A}$ the values $(r, \epsilon, \sigma, \gamma, a, d)$. This happens with negligible probability under the soundness property of ZkPoK, which has been proven in Lemma 1. During an HIDE event, $\mathcal{S}$ represents the dishonest VS to $\mathcal{T}$ and represents the honest customer/VS to $\mathcal{A}$. The simulation fails if $\mathcal{A}$ can break the the correctness of the t out of N verifiable secret sharing [28]. This happens with negligible probability. During a TRANS event, $\mathcal{S}$ represents the dishonest VS to $\mathcal{T}$ and represents the honest VS to $\mathcal{A}$. The simulation fails if $\mathcal{A}$ can break the the correctness of verifiable secret redistribution [29]. This happens with negligible probability.

V. Performance Evaluation

In this section, we evaluate the performance of DAPA. Since the core component of DAPA is PPIM, we mainly simulate PPIM in terms of computational costs and communication overheads. The simulation is implemented on a Macbook Pro notebook with Intel Core i7 processor. The memory is 8 GB and the clock rate is 2.6 GHz. For the cryptographic settings, we utilize the Java pairing based cryptography (JPBC) library [31] and choose the type F pairing (Barreto-Naehrig curve) in our simulation. To guarantee the least security level of 128 bits, the security parameters are set as $l = 1024$ and $\tilde{l} = 256$.

A. Computational Costs

To evaluate the computational costs, we first analyze the computational complexity of PPIM's each part (except *PGen* since it only needs to be run one time in the beginning). To measure the complexity, we count the number of time-consuming operations like public-key encryption, bilinear pairing, exponentiation in Z_{n^2}, $\tilde{G}$, $\mathfrak{G}$, and G_T. The computational delays of public-key encryption and bilinear pairing are denoted by PKE_T and $PAIR_T$, respectively. The computational delays of exponentiation in Z_{n^2}, $\tilde{G}$, $\mathfrak{G}$, and G_T are denoted by EXP_T^Z, $EXP_T^{\tilde{G}}$, $EXP_T^{\mathfrak{G}}$, and $EXP_T^{G_T}$, respectively. Assuming that there are N ($t = \lfloor \frac{N}{2} \rfloor + 1$) current committee members (VSs), $\hat{N}$ ($\hat{t} = \lfloor \frac{\hat{N}}{2} \rfloor + 1$) committee members in the next epoch, and χ engaged customers (means the customer who rents a shared car but does not return it), the results are shown in Table II. *PPIM.IDHide* is the major computational burden for the customer. The customer's computational complexity is $O(N)$ since the customer needs to split and distribute the secrets (y_1, y_2, y_3) according to the number of current committee members. Nevertheless, this calculation only needs to be performed once when the customer wants to rent a shared car. *PPIM.IDTransfer* is the major computational burden for VS. The computational burden of VSs should be discussed separately. The computational complexity of current committee members is $O(\chi \cdot \hat{t})$ since they have to transfer χ engaged customers' secrets to $\hat{N}$ next committee members with the threshold $\hat{t}$. The computational complexity of next committee members is $O(\chi \cdot N)$ since they need to verify the received χ secrets from N current committee members.

To demonstrate the computational efficiency of PPIM, we compare PPIM with four existing schemes, GSig-1 [32], GSig-2 [33], VEGS [34] and Vote-to-Link [35]. GSig-1 and GSig-2 are traditional group signature schemes where a customer can authenticate herself to CSSP using a group signature. CSSP can verify that the customer is a valid customer without knowing her identity and a centralized group manager can help trace the identity of the customer using a master secret key. VEGS is an adaptive group-signature-based scheme: a customer can encrypt her group signature and prove to CSSP that the uploaded

TABLE II
COMPUTATIONAL COMPLEXITY OF PPIM

PPIM	CSSP	Customer	VS
IDRegister	$EXP_T^{\tilde{G}}$	$2 * PAIR_T + EXP_T^{\mathfrak{G}}$	-
IDHide	-	$(12+3N+3t) * EXP_T^{Z} + 23 * EXP_T^{\tilde{G}} + 6 * EXP_T^{G_T} + 6 * PAIR_T$	$11 * EXP_T^{Z} + 17 * EXP_T^{\tilde{G}} + 9 * Pair_T + 9 * EXP_T^{G_T}$
IDTransfer	-	-	$\chi(3\hat{t} * EXP_T^{Z} + (3N+1) * EXP_T^{Z})$
IDRecover	$(3t+3) * EXP_T^{Z}$	-	PKE_T
IDRevoke	EXP_T^{Z}	-	-

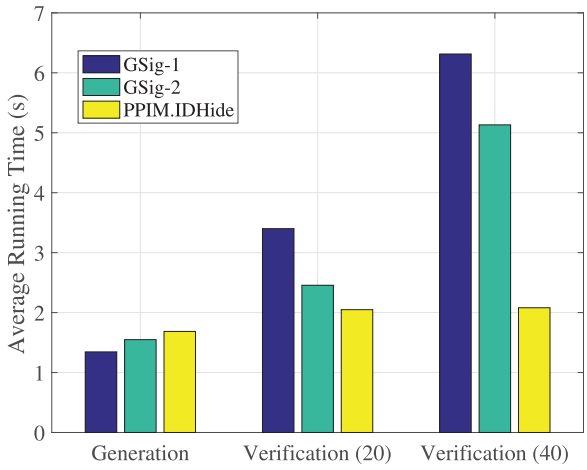


Fig. 2. Comparison with group signature schemes in terms of computational costs.

ciphertext containing a valid group signature can be verified and opened by an adjudicator (VEGS can be easily extended to support distributed adjudicators). Under the circumstances, the encrypted group signatures can be maintained by the adjudicator. Vote-to-Link is a threshold group-signature-based scheme where the customer can encrypt her identity credential using a threshold encryption scheme and prove to CSSP that the corresponding ciphertext contains a valid credential. In this case, the encrypted credential can be managed by multiple distributed moderators.

We first compare the computational delay of *PPIM.IDHide* with GSig-1 and GSig-2. To make the comparison fair, only one validation server is employed in PPIM and one group manager is employed in GSig-1 and GSig-2. Two main procedures are measured: signature generation and signature verification. The numbers of revoked customers are set as 20 and 40. The comparison results are shown in Fig. 2. GSig-1 and GSig-2 are more efficient than *PPIM.IDHide* in the generation of group signatures (anonymous identity credentials). The reason is that GSig-1 and GSig-2 utilize a lightweight linear encryption with chosen-plaintext security [30] while *PPIM.IDHide* utilizes a verifiable encryption with chosen-ciphertext security [25]. When CSSP verifies the signature, GSig-1 and GSig-2 requires CSSP to check revoked customers one by one. Hence, the verification is less efficient. Since *PPIM.IDHide* utilizes an accumulator technique [36], the costs of signature verification are constant with the increase of revoked customers. Second, we compare the computational delay of *PPIM.IDHide* with VEGS and Vote-to-Link. The performance metric is the computational delay of *PPIM.IDHide* with the different number of VSs. To make the comparison fair, we modify VEGS and Vote-to-Link to support the dynamic characteristic (i.e., the private key used for decrypting the identity credential is generated by the customer and is distributed to the VS (or adjudicator or moderator) dynamically). The comparison results are shown in Fig. 3(a) and 3(b). Compared with VEGS and Vote-to-Link, *PPIM.IDHide* is more efficient in terms of computational costs. VEGS is designed in the bilinear group of a composite order, and its computational cost is larger than *PPIM.IDHide* which is designed in the bilinear group of a prime order. Vote-to-Link uses an Elgamal-like threshold encryption method and the non-interactive zero-knowledge proof technique that cause more computational delay.

In addition, we also measure the computational delay of *PPIM.IDTransfer* with the different number of VSs. The results are shown in Fig. 3(c). To transfer one engaged customer' secret to the next committee, each current committee member only needs to perform the exponentiation in Z_{n^2}, which is efficient compared with other complex group operations. When the size of next committee is as large as 100, it takes around 200 ms to accomplish the transfer. Similarly, to receive one engaged customer' secret, each next committee member just needs to perform the exponentiation in Z_{n^2} as well. Considering that the size of current committee is as large as 100, it takes less than 1,200 ms to accomplish the verification, which is efficient. Note that, the delay can be optimized by increasing the epoch length since the frequency of identity transfer decreases. Also, we utilize Java programming language and the single-thread setting to simulate the procedure, and the delay can be further reduced by applying C programming language (or other low-level languages) and the multi-thread setting.

B. Communication Overheads

PPIM's communication overheads are mostly related to the number of distributed VSs and the number of engaged customers. If there exist a large number of committee members (VSs), more secrets are needed to be distributed to these VSs when the customer runs *PPIM.IDHide*. When the rental duration is long (e.g., several epochs), the current committee members need to transfer the managed secrets owned by engaged customers to next committee members. Therefore, we analyze the communication overheads of *PPIM.IDHide* and *PPIM.IDTransfer* as follows. Assuming that the sizes of $\tilde{G}$, G_T, and Z_{n^2} are denoted by $Size_{\tilde{G}}$, $Size_{G_T}$, and $4\,l$, the communication overhead of *PPIM.IDHide* between the customer and the current committee member is $(N+t+8) * 4l + (N+n') * \tilde{l} + 13 * Size_{\tilde{G}} + 2 * Size_{G_T}$. The customer does

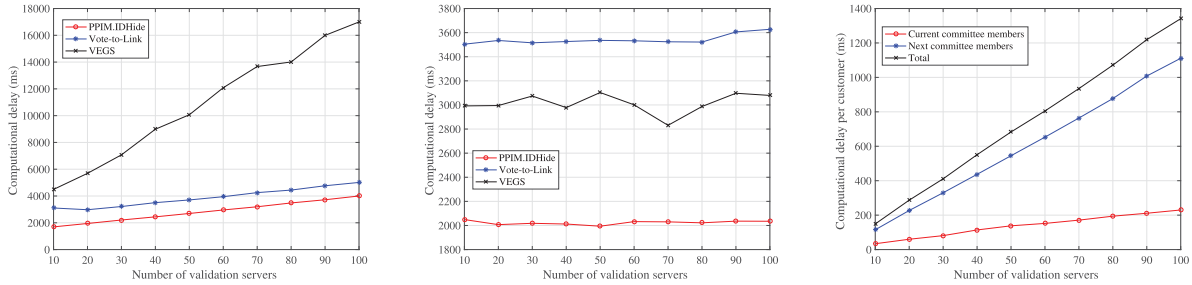


Fig. 3. Computational delay of customers and validation servers in each epoch (*PPIM.IDHide* and *PPIM.IDTransfer*). (a) Customer *(PPIM.IDHide)*. (b) Validation Server *(PPIM.IDHide)*. (c) Validation Server *(PPIM.IDTransfer)*.

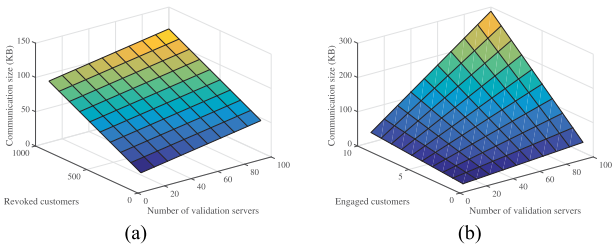


Fig. 4. Communication overheads of customers and validation servers (epoch). (a) Between customers and current committee members *(PPIM.IDHide)*. (b) Between current and next committee members *(PPIM.IDTranfer)*.

not only need to upload the public key (Y_1, Y_2, Y_3) to, share the secret shadow $\{s_i\}_{i=1}^{N}$ with, submit the encrypted secrets (u, w, v) and the proof π to the committee members but also needs to download the latest accumulator c and the revocation list S_{inv} from the bulletin board. The communication overhead of *PPIM.IDTransfer* between current committee members and next committee members is $\chi((\hat{t}-1)*4l+\hat{N}*\tilde{l})$. For each secret s_i, each current committee member P_i needs to write the commitment of the share $\hat{C}_{i,k}$ into the bulletin board and also share the secret shadow $\hat{s}_{i,j}$ privately with each committee member P_j in the next epoch. We show the effects of different numbers of VSs, revoked customers and engaged customers on the communication overhead in Fig. 4. Although the communication overheads are linear to the numbers of distributed VSs and engaged customers, it is still acceptable (less than 300 KB).

VI. Related Work

In this section, we review the recent works related to the security and privacy of car sharing services and discuss existing methods to resolve the conflict between privacy and accountability in the car sharing scenario.

A. Secure Car Sharing

Few works aim to solve the security and privacy issues in car sharing services [15], [16], [37], and they focus on the free-floating car sharing system but not the station-based car sharing system. Symeonidis *et al.* [37] proposed the first physical keyless car sharing system (KSS) where customers can share their cars with others remotely using a smartphone. They defined a threat model for car owners and customers, and also performed a security and privacy analysis of KSS. Then, they proposed a secure and privacy-enhancing scheme, named SePCAR [15], to address these threats. SePCAR provides generation and distribution of car access tokens for car sharing service, as well as update and revocation operations. To advance forensic evidence provision in the case of emergency, they applied a technique called secure multi-party computation (SMPC) [38], i.e., SePCAR utilized SMPC to achieve accountability while protecting customers' privacy. However, this method sacrifices computational efficiency since SMPC is time-consuming. Moreover, the requisite driving qualification checking is not mentioned in their scheme before customers share their cars. Similarly, Dmitrienko *et al.* [16] proposed a secure free-floating car sharing system that supports car sharing between customer and the car sharing service provider. The proposed system mainly focuses on an access control issue and is designed based on a two-factor authentication scheme including mobile devices and RFID tags. They did not consider the privacy of customers and thus a fully trusted car sharing service provider exists to manage the master keys for customers and vehicles.

B. Privacy & Accountability for Car Sharing

There are two types of traceable anonymous mechanisms which can be applied in the car sharing scenario to achieve both privacy preservation and accountability, pseudonym-based approaches [39]–[41] and group-signature-based approaches [32]–[35]. We focus on the group-signature-based approaches, where a customer can hide her real identity within a group by creating an indistinguishable (encrypted) signature to preserve her privacy. Compared with the pseudonym-based approaches, the group-signature-based approaches are more flexible and do not need complex pseudonym management and changing. Traditional group-signature-based approaches [32], [33] rely on a centralized group manager, which cannot be compromised at any time and should be always online with full trust. The centralized group manager can assist in generating anonymous identity credentials for customers and can trace an anonymous identity credential using a master secret key to determine which group member owns it. To overcome the shortages like the single point of failure, some recent group-signature-based approaches [34], [35] replace the centralized group manager with distributed authorities, i.e., only if a majority of distributed authorities agree,

TABLE III
COMPARISON OF PPIM WITH EXISTING SCHEMES

Properties	DST	DYN	CON	VER	REC	RVK
PPIM	✓	✓	✓	✓	✓	✓
GSig-1 [32]	✗	✗	✓	✓	✓	✓
GSig-2 [33]	✗	✗	✓	✓	✓	✓
VEGS [34]	✓	✗	✓	✓	✓	✗
Vote-to-Link [35]	✓	✗	✓	✓	✗	✗

DST: Distributed, DYN: Dynamic, CON: Confidential, VER: Verifiable, REC: Recoverable, RVK: Revocable

the real identity of a customer can be traced. Nevertheless, these authorities are considered to be fixed and stationary, and an adversary still has the chance to gradually compromise them in the car sharing scenario.

Different from the existing group-signature-based approaches [32]–[35], we propose a privacy-preserving identity management (PPIM) scheme for DAPA, where the authorities (a.k.a validation servers in DAPA) are periodically substituted to reduce the risk of being compromised. A customer cannot directly generate the group signature using validation servers' public keys as usual since they are dynamic. In other words, if a customer generates a group signature according to the public keys of the current validation servers, misbehaving customers cannot be traced after validation servers are offline and are substituted by another validation servers, which obeys our design goals. To show the difference between PPIM and the existing group-signature-based schemes [32]–[35], several properties are compared in Table III. DST denotes that there exist distributed validation servers. DYN denotes that the validation servers are dynamically substituted. CON denotes that an identity credential is encrypted as a ciphertext which cannot be distinguished. VER denotes that the identity credential can be verified by the validation servers without decryptions. REC denotes that the identity credential can be recovered. RVK denotes that the identity credential can be revoked.

VII. CONCLUSION

In this paper, we have proposed a decentralized, accountable, and privacy-preserving architecture for car sharing services (DAPA). In DAPA, decentralized and dynamic validation servers are employed to assist in managing customers' real identities instead of a single trusted authority, which significantly reduces the risk of the single point of failure and builds decentralized trust for customers. Meanwhile, based on a new privacy-preserving identity management scheme (PPIM), DAPA achieves privacy preservation for customers and accountability for car sharing service providers simultaneously. DAPA enables a car sharing service provider to verify the validity of customers' identifications without revealing customers' real identities, and it also allows a car sharing service provider to trace misbehaving customers no matter how validation servers change over time. Moreover, according to our experimental results, DAPA is efficient in terms of computational costs and communication overheads. In future work, we aim to design a lightweight car sharing architecture using smart contracts under the blockchain-based model, which achieves fairness among customers, car sharing service providers, and validation servers.

REFERENCES

[1] F. Ferrero, G. Perboli, M. Rosano, and A. Vesco, "Car-sharing services: An annotated review," *Sustain. Cities Soc.*, vol. 37, pp. 501–518, 2018.

[2] S. Weikl and K. Bogenberger, "Relocation strategies and algorithms for free-floating car sharing systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 4, pp. 100–111, Winter 2013.

[3] G. Wielinski, M. Trépanier, and C. Morency, "Carsharing service adoption in a dual-mode setting: A station-based and free-floating case study," Tech. Rep. 18-05472, 2018.

[4] F. Dandl and K. Bogenberger, "Comparing future autonomous electric taxis with an existing free-floating carsharing system," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 6, pp. 2037–2047, 2019.

[5] L. Hall, "The environmental benefits of car sharing," 2018. [Online]. Available: https://www.fleetcarma.com/environmental-benefits-car-sharing/. Accessed Jan. 15, 2019.

[6] C. Teale, "Zipcar: Each shared vehicle eliminates need for 13 personal vehicles," 2019. [Online]. Available: https://www.smartcitiesdive.com/news/zipcar-car-sharing-impact-report/546177/ Accessed Jan. 17, 2019.

[7] N. Fearn, "Car rental companies failing to protect customer data, claims privacy international," 2017. [Online]. Available: https://www.v3.co.uk/v3-uk/news/3022575/rental-companies-failing-to-protect-customer-data-says-report. Accessed Jan. 2, 2019.

[8] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "BlockChain: A distributed solution to automotive security and privacy," *IEEE Commun. Mag.*, vol. 55, no. 12, pp. 119–125, Dec. 2017.

[9] Y. Li, Q. Luo, J. Liu, H. Guo, and N. Kato, "TSP security in intelligent and connected vehicles: Challenges and solutions," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 125–131, 2019.

[10] C. Huang, R. Lu, X. Lin, and X. Shen, "Secure automated valet parking: A privacy-preserving reservation scheme for autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11 169–11 180, Nov. 2018.

[11] 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.*. to be published, doi: 10.1109/TDSC.2018.2850780

[12] J. Ni, X. Lin, and X. Shen, "Towards privacy-preserving valet parking in autonomous driving era," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 2893–2905, Mar. 2019.

[13] M. Hadian, T. Altuwaiyan, X. Liang, and H. Zhu, "Privacy-preserving task scheduling for time-sharing services of autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5260–5270, Jun. 2019.

[14] M. Langheinrich, "The golden age of privacy?" *IEEE Pervasive Comput.*, vol. 17, no. 4, pp. 4–8, Oct.–Dec. 2018.

[15] 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. Eur. Symp. Res. Comput. Secur.*, 2017, pp. 475–493.

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

[17] A. Madhusudan, I. Symeonidis, M. A. Mustafa, R. Zhang, and B. Preneel, "SC2Share: Smart contract for secure car sharing," in *Proc. Int. Conf. Inf. Syst. Secur. Privacy*, 2019, pp. 163–171.

[18] A. C. Hernandez, J. C.-Roca, and A. Viejo, "Key management system for private car-sharing scenarios," in *Proc. IEEE Veh. Technol. Conf.*, 2018, pp. 1–7.

[19] I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Proc. Annu. Int. Cryptology Conf.*, 2012, pp. 643–662.

[20] A. R. Choudhuri, M. Green, A. Jain, G. Kaptchuk, and I. Miers, "Fairness in an unfair world: Fair multiparty computation from public bulletin boards," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2017, pp. 719–728.

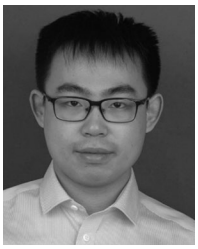
[21] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *J. Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[22] M. Z. Lee, A. M. Dunn, J. Katz, B. Waters, and E. Witchel, "Anon-pass: Practical anonymous subscriptions," *IEEE Secur. Privacy*, vol. 12, no. 3, pp. 20–27, May/Jun. 2014.

[23] S. Agrawal, C. Ganesh, and P. Mohassel, "Non-interactive zero-knowledge proofs for composite statements," in *Proc. Annu. Int. Cryptology Conf.*, 2018, pp. 643–673.

[24] E. Ben-Sasson *et al.*, "Zerocash: Decentralized anonymous payments from bitcoin," in *Proc. IEEE Symp. Secur. Privacy*, 2014, pp. 459–474.

[25] J. Camenisch and V. Shoup, "Practical verifiable encryption and decryption of discrete logarithms," in *Proc. Annu. Int. Cryptology Conf.*, 2003, pp. 126–144.
[26] M. H. Au, A. Kapadia, and W. Susilo, "BLACR: TTP-free blacklistable anonymous credentials with reputation," in *Proc. Annu. Netw. Distrib. Syst. Secur. Symp.*, 2012, pp. 1–17.
[27] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The second-generation onion router," in *Proc. USENIX Secur. Symp.*, 2004, pp. 303–320.
[28] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic," in *Proc. Annu. Int. Cryptology Conf.*, 1999, pp. 148–164.
[29] T. M. Wong, C. Wang, and J. M. Wing, "Verifiable secret redistribution for archive system," in *Proc. IEEE Secur. Storage Workshop*, 2002, pp. 94–106.
[30] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proc. Annu. Int. Cryptology Conf.*, 2004, pp. 41–55.
[31] A. De Caro and V. Iovino, "jPBC: Java pairing based cryptography," in *Proc. IEEE Symp. Comput. Commun.*, 2011, pp. 850–855.
[32] J. Shen, T. Zhou, X. Chen, J. Li, and W. Susilo, "Anonymous and traceable group data sharing in cloud computing," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 4, pp. 912–925, 2018.
[33] A. Ishida, Y. Sakai, K. Emura, G. Hanaoka, and K. Tanaka, "Fully anonymous group signature with verifier-local revocation," in *Proc. Int. Conf. Secur. Cryptography Netw.*, 2018, pp. 23–42.
[34] Z. Wang, X. Luo, and Q. Wu, "Verifiably encrypted group signatures," in *Proc. Int. Conf. Provable Secur.*, 2017, pp. 107–126.
[35] W. Lueks, M. H. Everts, and J. Hoepman, "Vote to link: Recovering from misbehaving anonymous users," in *Proc. ACM Workshop Privacy Electron. Soc.*, 2016, pp. 111–122.
[36] M. H. Au, P. P. Tsang, W. Susilo, and Y. Mu, "Dynamic universal accumulators for DDH groups and their application to attribute-based anonymous credential systems," in *Proc. Cryptographers' Track at RSA Conf.*, 2009, pp. 295–308.
[37] I. Symeonidis, M. A. Mustafa, and B. Preneel, "Keyless car sharing system: A security and privacy analysis," in *Proc. IEEE Int. Smart Cities Conf.*, 2016, pp. 1–7.
[38] P. Ananth, A. R. Choudhuri, A. Goel, and A. Jain, "Round-optimal secure multiparty computation with honest majority," in *Proc. Annu. Int. Cryptology Conf.*, 2018, pp. 395–424.
[39] A. Boualouache, S. Senouci, and S. Moussaoui, "PRIVANET: An efficient pseudonym changing and management framework for vehicular ad-hoc networks," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2019.2924856.
[40] Z. Liu, L. Zhang, W. Ni, and I. Collings, "Uncoordinated pseudonym changes for privacy preserving in distributed networks," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1465–1477, Jun. 1 2020.
[41] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in vanets," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 86–96, Jan. 2012.



Cheng Huang (Student Member, IEEE) received the B.Eng. and M.Eng. from Xidian University, Xi'an, China, in 2013 and 2016, respectively. He is currently working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. He was a Project Officer with the INFINITUS laboratory, School of Electrical and Electronic Engineering, Nanyang Technological University till July 2016. His research interests include applied cryptography, cyber security, and privacy in the mobile network.

Rongxing Lu (Senior Member, IEEE) received the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2012. He was an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from 2013 to 2016. He has been an Associate Professor with the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Fredericton, NB, Canada, since 2016. He was a Postdoctoral Fellow with the University of Waterloo, from 2012 to 2013. His research interests include applied cryptography, privacy enhancing technologies, and IoT-Big Data security and privacy. He was the recipient of the Governor Generals Gold Medal for his Ph.D. degree from the Department of Electrical and Computer Engineering, University of Waterloo, 8th IEEE Communications Society (ComSoc) AsiaPacific Outstanding Young Researcher Award in 2013, and the 2016 to 2017 Excellence in Teaching Award from FCS, UNB. He is currently the Vice Chair of IEEE ComSoc CIS-TC.

Jianbing Ni (Member, IEEE) received the B.E. degree and the M.S. degree from the University of Electronic Science and Technology of China, Chengdu, China, in 2011 and 2014, respectively. He received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2018. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON, Canada. His current research interests include applied cryptography and network security, with a focus on cloud computing, smart grid, mobile crowdsensing, and the Internet of Things.

Xuemin Shen (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is currently a Professor with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on network resource management, wireless network security, Internet of Things, 5G and beyond, and vehicular ad hoc and sensor networks. Dr. Shen is an Engineering Institute of Canada Fellow, a Canadian Academy of Engineering Fellow, a Royal Society of Canada Fellow, a Chinese Academy of Engineering Foreign Fellow, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and Communications Society. He is a Registered Professional Engineer of Ontario, Canada. He was the recipient of the R.A. Fessenden Award in 2019 from IEEE, Canada, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015 and Education Award in 2017 from the IEEE Communications Society. He was also the receipient of the Excellent Graduate Supervision Award in 2006 and Outstanding Performance Award five times from the University of Waterloo and the Premier's Research Excellence Award in 2003 from the Province of Ontario, Canada. He was the Technical Program Committee Chair/Co-Chair for the IEEE Globecom 2016, the IEEE Infocom 2014, the IEEE VTC 2010 Fall, the IEEE Globecom 2007, the Symposia Chair for the IEEE ICC 2010, and the Chair for the IEEE Communications Society Technical Committee on Wireless Communications. He was the Editor-in-Chief of the IEEE INTERNET OF THINGS JOURNAL and IEEE NETWORK, and the Vice President on Publications of the IEEE Communications Society.