Effective Privacy-Preserving Mechanisms for Vehicle-to-Everything Services

by

Cheng Huang

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2020

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner	NAME	Kui Ren
	Title	Professor
Supervisor	NAME	Xuemin (Sherman) Shen
	Title	University Professor
Internal Member	NAME	Xiaodong Lin
	Title	Associate Professor
Internal Member	NAME	Sagar Naik
	Title	Professor
Internal–external Member	NAME	Jun Liu
	Title	Associate Professor

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Owing to the advancement of wireless communication technologies, drivers can rely on smart connected vehicles to communicate with each other, roadside units, pedestrians, and remote service providers to enjoy a large amount of vehicle-to-everything (V2X) services, including navigation, parking, ride hailing, and car sharing. These V2X services provide different functions for bettering travel experiences, which have a bunch of benefits. In the real world, even without smart connected vehicles, drivers as users can utilize their smartphones and mobile applications to access V2X services and connect their smartphones to vehicles through some interfaces, e.g., IOS Carplay and Android Auto. In this way, they can still enjoy V2X services through modern car infotainment systems installed on vehicles.

Most of the V2X services are data-centric and data-intensive, i.e., users have to upload personal data to a remote service provider, and the service provider can continuously collect a user's data and offer personalized services. However, the data acquired from users may include users' sensitive information, which may expose user privacy and cause serious consequences. To protect user privacy, a basic privacy-preserving mechanism, i.e, anonymization, can be applied in V2X services. Nevertheless, a big obstacle arises as well: user anonymization may affect V2X services' availability. As users become anonymous, users may behave selfishly and maliciously to break the functions of a V2X service without being detected and the service may become unavailable. In short, there exist a conflict between privacy and availability, which is caused by different requirements of users and service providers. In this thesis, we have identified three major conflicts between privacy and availability for V2X services: privacy vs. linkability, privacy vs. accountability, privacy vs. reliability, and then have proposed and designed three privacy-preserving mechanisms to resolve these conflicts.

Firstly, the thesis investigates the conflict between privacy and linkability in an automated valet parking (AVP) service, where users can reserve a parking slot for their vehicles such that vehicles can achieve automated valet parking. As an optional privacy-preserving measure, users can choose to anonymize their identities when booking a parking slot for their vehicles. In this way, although user privacy is protected by anonymization, malicious users can repeatedly send parking reservation requests to a parking service provider to make the system unavailable (i.e., "Double-Reservation Attack"). Aiming at this conflict, a security model is given in the thesis to clearly define necessary privacy requirements and potential attacks in an AVP system, and then a privacy-preserving reservation scheme has been proposed based on BBS+ signature and zero-knowledge proof. In the proposed scheme, users can keep anonymous since users only utilize a one-time unlinkable token generated from his/her anonymous credential to achieve parking reservations. In the mean-

time, by utilizing proxy re-signature, the scheme can also guarantee that one user can only have one token at a time to resist against "Double-Reservation Attack".

Secondly, the thesis investigates the conflict between privacy and accountability in a car sharing service, where users can conveniently rent a shared car without human intervention. One basic demand for car sharing service is to check the user's identity to determine his/her validity and enable the user to be accountable if he/she did improper behavior. If the service provider allows users to hide their identities and achieve anonymization to protect user privacy, naturally the car sharing service is unavailable. Aiming at this conflict, a decentralized, privacy-preserving, and accountable car sharing architecture has been proposed in the thesis, where multiple dynamic validation servers are employed to build decentralized trust for users. Under this architecture, the thesis proposes a privacy-preserving identity management scheme to assist in managing users' identities in a dynamic manner based on a verifiable secret sharing/redistribution technique, i.e. the validation servers who manage users' identities are dynamically changed with the time advancing. Moreover, the scheme enables a majority of dynamic validation servers to recover the misbehaving users' identities and guarantees that honest users' identities are confidential to achieve privacy preservation and accountability at the same time.

Thirdly, the thesis investigates the conflict between privacy and reliability in a road condition monitoring service, where users can report road conditions to a monitoring service provider to help construct a live map based on crowdsourcing. Usually, a reputation-based mechanism is applied in the service to measure a user's reliability. However, this mechanism cannot be easily integrated with a privacy-preserving mechanism based on user anonymization. When users are anonymous, they can upload arbitrary reports to destroy the service quality and make the service unavailable. Aiming at this conflict, a privacy-preserving crowdsourcing-based road condition monitoring scheme has been proposed in the thesis. By leveraging homomorphic commitments and PS signature, the scheme supports anonymous user reputation management without the assistance of any third-party authority. Furthermore, the thesis proposes several zero-knowledge proof protocols to ensure that a user can keep anonymous and unlinkable but a monitoring service provider can still judge the reliability of this user's report through his/her reputation score.

To sum up, with more attention being paid to privacy issues, how to protect user privacy for V2X services becomes more significant. The thesis proposes three effective privacy-preserving mechanisms for V2X services, which resolve the conflict between privacy and availability and can be conveniently integrated into current V2X applications since no trusted third party authority is required. The proposed approaches should be valuable for achieving practical privacy preservation in V2X services.

Acknowledgements

The past four years of my Ph.D. study at the University of Waterloo are truly the most memorable, precious and significative time in my life. I would like to thank all the people who greatly support my Ph.D. study. They are my supervisor, my thesis committee members, my colleagues, and my families. Without their help and encouragement, I would not have such research achievements and enjoy the PhD study period.

First of all, I would like to express my heartfelt gratitude to my supervisor, Professor Xuemin (Sherman) Shen, who always patiently guides me, encourages me, and gives many valuable suggestions to make me successfully complete this thesis. He not only helps me develop the academic skills, but also teach me a lot of useful life experiences. I am really inspired by his dedication and enthusiasm to his work, his students and his family. I also gratefully acknowledge Professor Rongxing Lu, for his great efforts to inspire my research ideas, discuss with me, and help me acquire the research achievements. In addition, I appreciate the honorable members of my thesis committee, Professor Kui Ren, Professor Sagar Naik, Professor Jun Liu, and Professor Xiaodong Lin. Their insightful comments have significantly affected the substance and presentation of my work.

During my Ph.D. study, I have always been working together with my colleagues in the BBCR lab. Although sometimes doing research is boring and daunting, my colleagues in BBCR group have made my life colorful and enjoyable. I would like to especially thank Professor Kuan Zhang, Professor Jianbing Ni, Professor Nan Cheng, Professor Yuan Zhang, Professor Anjia Yang, Professor Haomiao Yang, Professor Qi Jiang, Professor Dajiang Chen, Professor Wei Wang, Professor Meng Li, Professor Wenjuan Tang, Dr. Nan Chen, Dr. Wenchao Xu, Dr. Wen Wu, Dr. Junlin Li, Dr. Weisen Shi, Dr. Yuanyuan He, Dongxiao Liu, Liang Xue, Hao Ren, Jialu Hao, Chuan Zhang, Fuyuan Song, Manaf Bin-Yahya and Jinwen Liang, for their inspiring discussions and invaluable insights on my research. I gratefully acknowledge all BBCR group members for their continuous encouragement, selfless help and all the good times we spent together.

There are many other people whose names are not mentioned due to the limited spaces. It does not mean that they are not important and I have forgotten or ignored their help. It is a privilege for me to work and share life with so many bright, energetic and helpful people.

The thesis is also dedicated to my father Wenyao Huang, my mother Jianying Chen, and my girlfriend Yali Fan. Their love and encouragement have been and will always be a great source of inspiration in my life. Thanks to them all for their continuous and ever-caring support which made me always feel their presence so near to me. I would continuingly work hard to fulfill my career goals and never disappoint them.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

V2X	Vehicle to Everthing
AV	Autonomous Vehicle
C-V2X	Celluar-based V2X
DSRC	Dedicated Rhort-Range Communication
QoS	Quality of Service
V2V	Vehicle to Vehicle
V2I	Vehicle to Infrastructure
V2P	Vehicle to Pedestrain
V2N	Vehicle to Network
UE	User Equipment
RAN	Radio Access Network
AES	Advanced Encryption Standard
BBS+	Boneh-Boyen-Shacham Signature Plus
PS	Pointcheval-Sanders Signature
DL	Discrete Logarithm
CDH	Computational Diffie-Hellman
DDH	Decisional Diffie-Hellman
1-FlexDH	1-Flexible Diffie-Hellman
q-SDH	q-Strong Diffie-Hellman
q-DDHI	q-Decisional Diffie-Hellman Inversion
RSA	Rivest-Shamir-Adleman
ZkPoK	Zero-knowledge Proof-of-Knowledge
NIZK	Non-interactive Zero-knowledge Proof
PPIM	Privacy-Preserving Identity Management

DAPA	Decentrialized, Accountable, and Privacy-Preserving Architecture
AVP	Automated Valet Parking
SM	Smartphone
SMPC	Secure Multi-Party Computation
PLT	Parking Lot Terminal
VS	Validation Server
GDPR	General Data Protection Regulation
PSP	Parking Service Provider
MSP	Monitoring Service Provider
CSSP	Car Sharing Service Provider
TA	Trusted Authority
TTPA	Trusted Third Party Authority
TLS	Transport Layer Security
AI	Artificial Intelligence
PKI	Public Key Infrastructure
JPBC	Java Paring Based Cryptograpy

Chapter 1

Introduction

With the advancement of wireless communication technologies, from dedicated short-range communication (DSRC) [1] and LTE-V2X [2] to 5G V2X [3], "Vehicle-to-Everything" (V2X) communication has become more critical and realistic, which paves the way for future intelligent transportation systems and fully automated driving applications. Through V2X communications, plenty of V2X applications can be enabled [4] to offer various V2X services to the drivers and the passengers, with the assistance of smart vehicles and smartphones, including parking, navigation, and ride hailing, which can lead to a more convenient driving experiences for drivers and more effective traffic management.

Although V2X services bring the benefits for both users and service companies, they also face one of the fundamental challenges, i.e., how to protect user's privacy [5, 6]. As the communication is not a bottleneck for V2X services, current V2X applications are more data-intensive and require users to upload a lot of personal data to help increase their service quality. On one hand, these data can be well utilized by the service companies in many aspects, e.g., building a more accurate artificial intelligence (AI) model for their services. On the other hand, these data may reveal users' sensitive information, such as locations, preferences, and habits [7]. In particular, the company, equipped with powerful computational resources, can even take the advantage of AI technologies to extract more sensitive information from a user's data through deep learning analysis, which would seriously violate the user privacy. Therefore, protecting user privacy is an imperative requirement from users' perspective. However, privacy preservation for V2X services is still in a quite embarrassing situation. While some privacy-preserving mechanisms can be applied in these services, most of them will severely affect the system availability and eventually may make these services unavailable. In this thesis, we investigate the privacy

issues of V2X services, especially focusing on identity privacy, and propose several effective privacy-enhancing mechanisms for V2X services.

1.1 V2X Communication and Its Services

Vehicle-to-everything (V2X) communication can be regard as an umbrella term of vehicle-related communication systems, which provides real-time and highly reliable information flows to enable different kinds of V2X services for drivers. Currently, two representative V2X communication modes are the dedicated short-range communication (DSRC) based on the IEEE 802.11p standard, and the cellular vehicle-to-everything communication (C-V2X) based on 3GPP LTE network. Although the DSRC has been proposed earlier than the C-V2X and commercialized for many years, it still has several inextricable limitations due to the vehicles' high mobility and high quality-of-service (QoS) provisioning requirements of V2X services. Alternatively, LTE C-V2X and its future 5G NR-based C-V2X are built on existing cellular infrastructures, which can provide better QoS support, low latency, larger coverage, higher and more reliable data rate for moving vehicles, paving the road to future connected vehicles and autonomous vehicles. Based on the completed 3GPP Release 16 specification [8], the C-V2X offers superior performance over the DSRC in terms of coverage, mobility support, transmission delay, reliability and scalability, which turns the C-V2X into the most suitable candidate meeting the requirements of V2X services.

1.1.1 V2X Communication Architecture

A general V2X communication architecture mainly consists of four entities [9]: user equipment (UE), radio access network (RAN), core network, and third-party service providers, as shown in Figure 1.1.

- ▷ User Equipment (UE): UE includes smart vehicles equipped with the on-board units and mobile smart devices (e.g., smartphones and tablets) carried by drivers and pedestrians. They can communicate with other UE through V2X communications.
- ▷ Radio Access Nework (RAN): RAN includes base stations (or radio transceivers or road-side units) which act as intermediates to connect UE to the core network. Most base stations are primarily connected via fiber backhaul to the core network. It also assists ib coordinating the management of resources across the base stations.

- ▷ Core Network: core network mainly provide networking services for UE accessed through RAN based on innovative networking technologies, such as software-defined networking and network function virtualization. It also connects to the external service providers such that various V2X services can be provided for UE.

- ▷ Service Providers: Service providers are different V2X service companies that maintain and offer different V2X services to UE such as parking, navigation and ride hailing.

There are mainly four types of V2X communications in this architecture, including vehicle-to-vehicle (V2V) communication, vehicle-to-pedestrian (V2P) communication, vehicle-to-infrastructure (V2I) communication and vehicle-to-network (V2N) communication.

- ▷ *V2V communication.* Vehicles can communicate with other vehicles or UE and exchange useful information through device-to-device communication/broadcasting or via the infrastructure. Namely, V2V communications allow UE to transmit messages carrying self-generated data to other vehicles, such as location and speed, road conditions, and traffic flow information. For example, a vehicle, after receiving a nearby vehicle's attribute-related data, can alert the current driver in case of a predictable collision with this vehicle in the same lane and direction [10].

- ▷ *V2P communication.* Similar to V2V communications, V2P communications expect the information exchange between pedestrians' UE and vehicular UE to minimize potential dangers and raise the acceptance of autonomous vehicles (AVs) on roads. For example, people walking, children being pushed in strollers, people using wheelchairs or bicyclists, can be detected by the moving vehicles to avoid dangerous situations. Meanwhile, the pedestrians can also be notified with their movements. Different from the vehicles, the pedestrians' UE normally have a lower battery capacity, therefore they may not be able to send/receive messages with the same periodicity [11].

- ▷ *V2I communication.* The UE supporting V2I communication transmits messages containing vehicle-related information to a local base station or road-side unit (RSU), and relevant local edge nodes. Generally, a local base station/RSU serves a particular geographic area and multiple base stations/RSUs can serve overlapping areas. For example, the traffic light device in the intersection can be a local edge node to communicate with the vehicles which are close to them, and adjust their red/green light periods accordingly [12].

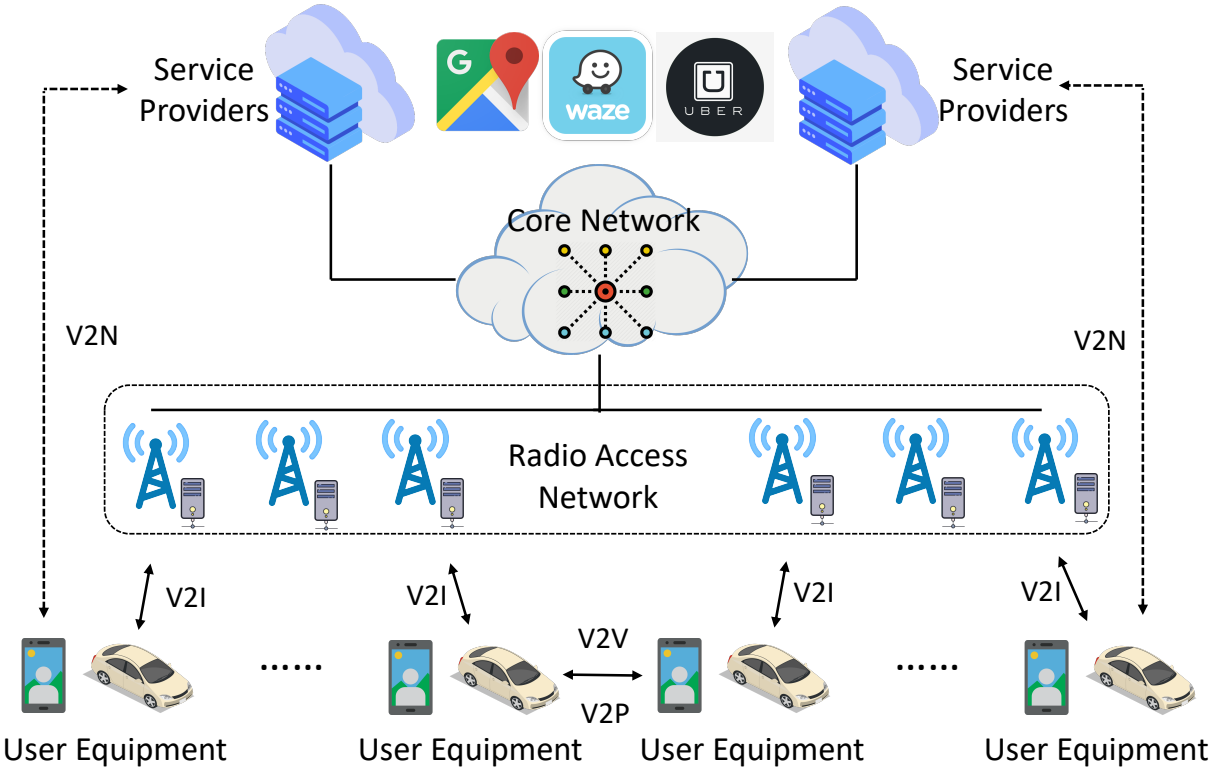


Figure 1.1: A general V2X communication architecture

▷ *V2N communication.* The UE can communicate with a remote service provider through V2N communications. Various applications can be deployed and are able to provide multiple services for the drivers and passengers. A fashionable case is the ride-hailing applications, e.g., Uber and Lyft, where the passengers can use the UE to hail the nearby vehicles remotely and the nearby drivers can pick the passengers up and ride the passengers to their desired locations [13].

1.1.2 V2X Services

Under the V2X communication architecture, we focus on V2X services that provided by remote service providers, which have been widely accepted. These services can be abstracted into a system that mainly involves two entities: remote service providers and users.

Remote service providers are companies that maintain various V2X services in online servers, and they also have connections with other entities related to vehicles, e.g., electric vehicle charging stations. Users including driver and passengers that would like to make use of V2X services provided by remote service providers, such as parking, navigation, and ride hailing. Since current smart connected vehicles are still in its developing and most of them are not equipped with V2X communication modules like antennas and on-board unit. In the real world, most of users rely on their smartphones to connect to these V2X services by installing mobile applications published by remote service providers. Users' smartphones can also connect to vehicles through some interfaces, such as IOS Carplay and Android Auto, and users are allowed to utilize these services in the car infotainment system.

Typical V2X services including navigation services, parking services, car sharing services, ride hailing services, road condition monitoring services, etc. These services have already been deployed by industrial companies, such as Google Map [1], Uber [2], WAZE [3], ZipCar [4], Impark[5]. Some real-world V2X services are introduced as follows, which bring a lot of convenience to our daily lives.

- ▷ Ride Hailing Services: In ride hailing services [13], users can recruit proper drivers to drive them to the destinations where they would like to go. The process of a ride hailing service is as follows. A user first sends a ride request to a ride hailing service provider using a ride hailing mobile application installed on the smartphone, and the service provider can match drivers with the request to locate a proper driver according to the location information. Then, the chosen driver will receive a request from the service provider. If the request is confirmed by the driver, he/she then drives to the user's place to pick up the user and drives to the appointed destination. If the request is denied by the driver, the service provider can find another driver until a driver confirms the request or a user withdraws a request. After the user reaches the destination, the user can pay for the ride-hailing service through the smartphone.

- ▷ Automated Valet Parking Services: In Automated Valet Parking (AVP) services [14], a user can utilize his/her smartphone to achieve parking without human intervention. The process of an AVP service is as follows, a user can first drop off his/her

[1] `https://www.google.com/maps`
[2] `https://www.uber.com/`
[3] `https://www.waze.com/`
[4] `https://www.zipcar.com/`
[5] `https://www.impark.com/`

autonomous vehicle at a specific dropping-off place, and use a parking mobile application installed on the smartphone to book a nearby parking slot by sending a parking reservation request to a parking service provider. Then, the parking service provider can lock a nearby unoccupied parking slot for the user and inform the user that a parking slot is prepared. Next, the user can enable his/her autonomous vehicle to automatically achieve the parking process, i.e., the vehicle can self-drive to the reserved parking slot. Finally, the user can pick up his/her vehicle by finishing the payment on his/her smartphone and enable the vehicle to self-drive to a specific pick-up position.

- ▷ Car Sharing Service: In car sharing services [15], a user can utilize his/her smartphone to rent a shared car without human intervention. The process of a car sharing service is as follows, a user can first reserve a shared car at a nearby car sharing station using a car sharing mobile application installed on the smartphone, by sending a reservation request to a car sharing service provider. After receiving the request, the service provider can lock the unoccupied vehicle at the car sharing station, and send a response back to the user. After receiving the response from the service provider, the user can go to the car sharing station to pick up the shared vehicle, unlock the vehicle using his/her smartphone, and drive anywhere. When a user would like to return the vehicle, the user can search and select a car sharing station nearby his/her destination, return the shared car at the chosen station, and pay for the car-sharing service using his/her smartphone.

- ▷ Crowdsourcing-based Road Condition Monitoring Services: Crowdsourcing-based road condition monitoring services [16] enable a user to report road conditions as voluntary mobile sensors using a monitoring mobile application installed on his/her smartphone. The services bring a lot of advantages compared with traditional road condition monitoring services based on fixed sensors, including improving the sensing coverage range, reducing the delay of local traffic perturbation updating, and reducing the costs of sensor deployment. After collecting the road conditions from users, a real-time map can be updated and shared with all users such that users can obtain changes of road conditions and traffic conditions timely. In the meantime, the real-time map can be utilized to improve the service quality of other V2X services, e.g. increasing the accuracy of navigation.

1.2 Privacy Requirements in V2X Services

Privacy has become a fundamental right in the digital age. The evolution of technologies brings people into a new world, where almost any data can be stored electronically for a long period and can be online and available. Under the circumstances, users, when using the third-party companies' services, need to upload personal information and cannot control their data in a proper way. Actually, one-time data breach is not so important since it cannot reveal much information related to one person, but a very dangerous and often ignored fact about privacy is that the data from one person can be accumulated over a long period and be evaluated automatically, that is, even small correlations of the data may reveal useful information. For instance, the Facebook-Cambridge Analytica data scandal involves the collection of personally identifiable information of up to 87 million Facebook users since 2014. The accumulated data is analyzed and used to attempt to influence voters' opinions on behalf of politicians who hired them. Long-term data analysis gives the chance for the Cambridge Analytica company to learn the private behaviors of users and leads them to its desired voting outcome. What's worse, once user privacy is lost, it is very hard to re-establish that state of personal rights. The situation also happens in V2X services. Diverse mobile devices owned by drivers, such as smart vehicles and smartphones highly increase the possibility that a person's data will be collected by accompanying V2X service providers and the risks of data disclosure. For example, Uber suffers from a crucial data breach, where a hacker breaks Uber servers to gain access to personal information of 57 million riders and drivers, including names and driver's license numbers for 600,000 drivers.

These privacy leakage events have attracted a lot of attentions, and many standards related to security and privacy in V2X services have been proposed. The IEEE 1609.2 standard [17] and the 3GPP TS 33.185 standard [18] both give the official illustration that user privacy is a necessary part of V2X services. From the perspective of laws, starting from 2017, the General Data Protection Regulation (GDPR) has harmonized data protection laws to protect user privacy. The GDPR imposes new rules on companies and organizations that offer services to people, whose primary objective is to give people back control of their personal data. That is, in this era of privacy protection, the companies does not only need to offer high-quality service to their customers, but also need to follow a privacy-by-design principle, address privacy concerns and comply with regulations especially regarding the GDPR. If serious data breaches happen under the GDPR, a company can be fined either 20 million euros or up to 4% of their annual revenues, which is a great amount. Besides, the researchers and industry communities also put a lot of efforts in this area. The University of Alberta in Canada has received approximately $500,000 in funding for a connected vehicle

privacy program [19], and the Privacy4Cars company [6], has published the first mobile app designed to help erase personally identifiable information from modern vehicles, which enables consumers and businesses to quickly and efficiently delete personal data retained by modern vehicle infotainment systems.

Privacy can be breached by both internal attackers and external attackers in V2X services. As internal attackers, V2X service providers cannot be fully trusted and may be curious about their users' data as there may exist malicious employees that will trade users' personal data for self-interests [20] if all user data is accessible. Other users cannot be trusted as well since some of them may be malicious to collude with the service provider to steal honest users' data. External attackers like hackers or competitors may also aim to compromise user privacy. Hackers can steal users' personal information for data selling in illegal markets while competitors may break other competitor's V2X services such that affected users may move to their services due to the data leakage events. According to the attacks, privacy requirements in V2X services can be roughly categorized into three classes: identity privacy, location privacy, and content privacy.

- ▷ Identity Privacy. Identity privacy normally means the anonymity of users, i.e., users cannot be identified from an anonymity set of users in V2X services. It is a basic privacy requirement for V2X services, since V2X services will continuously collect a user's sensitive data and the accumulated data of one user may reveal sensitive information of a user, such as his/her home location, his/her preferences about point of interests, his/her career, etc. Specifically, anonymity can be further divided into two properties: pseudonymity and unlinkability. Pseudonymity means a user uses pseudonyms as specific identifiers when communicating with others. Unlinkability means the pseudonyms, belonging to one user, cannot be linked by others, that is, two pseudonyms cannot be exactly identified from the same person even if they comes from the same person.

- ▷ Location Privacy. Location privacy is a privacy requirement built based on identity privacy but with some extensions. In some situations, anonymity is enough to provide location privacy protections for users, i.e., anonymous users reveal their location information to a V2X service provider, but the V2X service provider cannot determine the sources of location information and link a user's two locations. However, in some extreme situations, e.g., only one user exists, or the V2X service provider can link the location of a user based on some background information, additional location privacy requirements are needed, i.e., users reveal their obfuscated or faked location

[6] `http://privacy4cars.com/`

information to a service provider, and a V2X service provider cannot determine the precise location information of users.

- ▷ Content Privacy. Different from identity privacy and location privacy, content privacy is a more broad concept, which normally means that the data content owned by users should be concealed before sharing with V2X services such that only authorized entities can obtain the data content. The data content may include users' identities and locations, but also include many other information such as reputation. Specifically, content hiding can be further divided into two types : 1) users can encrypt the content with an encryption key, and others without a decryption key cannot decrypt the ciphertext to obtain the content; and 2) users can perturb or obfuscate the data with noises, and others cannot extract the original data from the noised content. From another perspective, content privacy also means that a proper fine-grained access control mechanism should be deployed to determine the access permission of the content.

1.3 Research Motivations and Objectives

The motivation of this thesis is to protect user privacy in V2X services. More specifically, the thesis focuses on protecting users' identity privacy through anonymization, which is a basic privacy requirement for preventing V2X services from collecting user behavior. Namely, when users enjoy V2X services provided by third-party companies, they can choose to hide their identities as an option. Note that, when anonymity is guaranteed, location privacy and content privacy are protected somehow, since the linkage among identity, location, content have been broken.

However, this basic privacy requirement may be seriously conflict with the availability of V2X services, since the privacy is always a double-edged sword. As users are anonymous, some users may perform malicious behavior that breaks the system functions of V2X service, and eventually makes the V2X services unavailable. Informally, for some V2X services that have access limitations for users, strong anonymity always means that unlimited access permissions, which may damage the system. We name the conflict as "privacy vs. linkability". For some V2X services that need to trace a user's real identity, strong anonymity means a user cannot be accountable, which may break the system. We name the conflict as "privacy vs. accountability". For some V2X services that need to judge a user' reliability according to the user's historical behavior, strong anonymity always means that the user does not have historical behavior, which may ruin the system. We

name the conflict as "privacy vs. reliability". In the following, we use three typical V2X services to clearly demonstrate these conflicts, including automated valet parking (AVP) services, car sharing services, and crowdsourcing-based road condition monitoring services.

- ▷ Privacy vs. Linkability: Linkability is required in an AVP service because it can assist a parking service provider in resisting against a special attack on the system availability, i.e., one user maliciously repeats a parking reservation request, occupies all nearby parking slots, and finally destroys the system. This attack does not exist if all users use real names to book parking slots, but users' identity privacy cannot be protected under this circumstance. If a strong anonymous mechanism has been applied in an AVP system and the parking service provider cannot link a user's two parking requests, the attack becomes a critical issue and the availability of the AVP service cannot be guaranteed. Hence, how to resolve the conflict between privacy and linkability is a very challenging issue.

- ▷ Privacy vs. Accountability: Accountability is required in a car sharing service. It is necessary since a car sharing service provider should have the ability to reveal a user's identity for liability issues in the real world, e.g., a user who maliciously damages a shared vehicle should be traced. If all users use real names to reserve shared vehicles, the accountability is naturally satisfied. If an anonymous mechanism has been applied in the car sharing system, the car sharing service provider cannot trace a user's real identity due to user anonymization. Accordingly, the availability of the car sharing service cannot be guaranteed. As a result, how to resolve the conflict between privacy and accountability is a very challenging issue.

- ▷ Privacy vs. Reliability: Reliability is required in a crowdsourcing-based road condition monitoring service. When a user makes a road condition report, a monitoring service provider should have the capability to judge the reliability of the report. Generally, the reliability is determined by the reputation score of the user who made the report and the reputation score is calculated based on the user's previous reports (historical information). If all users use real names to report, the monitoring service provider can easily link a life-time reputation score to each user and the reliability of every report can be measured. In contrast, if a strong anonymous mechanism has been applied. the monitoring service provider cannot bind a user's real identity with an updatable reputation score due to user anonymization. Accordingly, the availability of the car sharing service cannot be guaranteed. Consequently, how to resolve the conflict between privacy and reliability is a very challenging issue.

In summary, the objective of this thesis is to resolve the conflict between privacy and availability, and to design effective privacy-preserving mechanisms for V2X services.

1.4 Research Contributions

To achieve the above-mentioned objectives, we develop a suite of privacy-preserving schemes. Specifically, the main contributions lie in the following aspects:

- *Resolving the conflict between privacy and linkability*: To resolve the conflict, we propose a new privacy-preserving parking reservation scheme for securing AVP systems. Specifically, each anonymous user must have only one valid reservation token at any moment, and the token can only be used for booking one vacant parking space once. The proposed scheme does not only preserve the user's identity privacy and location privacy but also prevents the "Double-Reservation Attack" based on several elegant building blocks, i.e., zero-knowledge proofs of knowledge and proxy re-signature. Detailed security analysis confirms the security properties of our proposed scheme. In addition, extensive simulations are conducted to compare our proposed scheme with three previous schemes, and the experiment results demonstrate that our scheme is also much efficient in a WiFi-based testbed.

- *Resolving the conflict between privacy and accountability*: To resolve the conflict, 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.

- *Resolving the conflict between privacy and reliability*: To resolve the conflict, we propose a privacy-preserving crowdsourcing-based road condition monitoring scheme, which innovatively supports anonymous user reputation management. Specifically, based on well-designed zero-knowledge proofs, a user can anonymously generate a road condition report with a hidden reputation score, while the monitoring service provider can verify the user's reputation score and authenticate the report in a privacy-preserving manner. By utilizing homomorphic commitments, the proposed scheme does not require any other third party to manage a user's reputation score, as the user can self-maintain it locally and get it updated with the help of the monitoring service provider according to the accuracy of his/her reports. Moreover, we also design a K-bound reputation updating mechanism such that a user needs to update his/her reputation score to keep it up-to-date after reporting at most K times. Detailed security analysis shows that the proposed scheme achieves three security properties: anonymity, K-tolerant trust, and unforgeability. In addition, a proof-of-concept prototype is developed based on JAVA, and performance evaluation via extensive simulations demonstrates the feasibility and practicality of the proposed scheme in terms of computational and communication overhead.

1.5 Thesis Outline

The remainder of this thesis is organized as follows: Chapter 2 reviews the preliminaries exploited to design schemes and introduces a comprehensive overview of literatures related to anonymous mechanisms and privacy-preserving schemes for V2X services. Chapter 3 develops a privacy-preserving scheme to resolve the conflict between privacy and linkability in an automated valet parking service. Chapter 4 investigates the conflict between privacy and accountability and proposes a new decentralized, privacy-preserving, and accountable architecture for car sharing services. Chapter 5 mitigates the conflict between privacy and reliability by proposing a privacy-preserving crowdsourcing-based road condition monitoring scheme that supports anonymous reputation management. Finally, Chapter 6 concludes the thesis, and introduces our future research directions.

Chapter 2

Background

This chapter introduces the background of privacy-preserving mechanisms for V2X services. We first review the underlying techniques leveraged to design the proposed privacy-preserving schemes. Then, we give a comprehensive survey on the literature of anonymous mechanisms and privacy-preserving mechanisms for V2X services.

2.1 Basic Techniques

We review the preliminaries, including number-theoretic problems, bilinear pairing, pseudo random function, Pedersen commitment, proxy re-signature, BBS+ signature, PS signature, dynamic accumulator, and zero-knowledge proof.

2.1.1 Number-Theoretic Problems

Many secure and privacy-preserving schemes are designed based on the intractability of solving some hard problems. The following problems are presented since they are relevant to this thesis. No probabilistic, polynomial time algorithm has non-negligible advantage in solving the following problems.

Discrete Logarithm (DL) assumption [21]. The DL problem in a prime-order group $\mathbb{G}$ is defined as follows: Given g^a, where g is a generator of $\mathbb{G}$, no adversary can extract a, in probabilistic polynomial time with non-negligible probability, then we say that the DL assumption in $\mathbb{G}$ holds.

Computational Diffie-Hellman (CDH) assumption [22]. The CDH problem in a prime-order group $\mathbb{G}$ is defined as follows: Given $(g, g^a, g^b) \in \mathbb{G}^3$, where g is a generator of $\mathbb{G}$, no adversary can compute g^{ab}, in probabilistic polynomial time with non-negligible probability, then we say that the CDH assumption in $\mathbb{G}$ holds.

Decisional Diffie-Hellman (DDH) assumption [22]. The DDH problem in a prime-order group $\mathbb{G}$ is defined as follows: Given $(\hat{g}, \hat{g}^a, \hat{g}^b, \hat{g}^c) \in \mathbb{G}^4$, where g is a generator of $\mathbb{G}$, no adversary can determine $c = ab$ or not, in probabilistic polynomial time with non-negligible probability, then we say that the DDH assumption in $\mathbb{G}$ holds.

q-Strong Diffie-Hellman (q-SDH) Assumption [23]. The q-SDH problem in a prime-order group $\mathbb{G}$ is defined as follows: Given a $(q+2)$ tuple $(g, g_0, g_0^x, g_0^{x^2}, \cdots, g_0^{x^q}) \in \mathbb{G}^{q+2}$, output a pair (A, c) such that $A^{(x+c)} = g_0$ where $c \in Z_p^*$. We say that the q-SDH assumption in $\mathbb{G}$ holds if there is no algorithm can solve the q-SDH problem in $\mathbb{G}$ with non-negligible advantage in probabilistic polynomial time.

q-Decisional Diffie-Hellman Inversion (q-DDHI) Assumption [24]. The q-Decisional Diffie-Hellman Inversion problem (q-DDHI) in a prime-order group $\mathbb{G}$ is defined as follow: On input a $(q+2)$-tuple $(g, g^x, g^{x^2}, ..., g^{x^q}, g^c) \in \mathbb{G}^{q+2}$, output 1 if $c = 1/x$ and 0 otherwise. We say that the q-DDHI assumption in $\mathbb{G}$ holds if there is no algorithm can solve the q-DDHI problem in $\mathbb{G}$ with non-negligible advantage in probabilistic polynomial time.

1-Flexible Diffie-Hellman Assumption [25]. The 1-Flexible Diffie-Hellman Assumption (1-FlexDH) in a prime-order group $\mathbb{G}$ is defined as follow: On input a 3-tuple $(g, A = g^a, B = g^b) \in \mathbb{G}^3$, no adversary can output a 3-tuple $(C, C^a, C^{ab}) \in (\mathbb{G} \backslash 1_{\mathbb{G}})^3$ in probabilistic polynomial time with non-negligible probability, then we say that the 1-FlexDH assumption in $\mathbb{G}$ holds.

PS assumption [26]. Let $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e)$ be a type-3 bilinear group. g is a generator of $\mathbb{G}_1$ and $\tilde{g}$ is a generator of $\mathbb{G}_2$. For $\tilde{X} = \tilde{g}^x$ and $\tilde{Y} = \tilde{g}^y$ where x and y are random elements chosen from Z_p, we define the oracle $O(m)$ on input $m \in Z_p$ that chooses a random $h \in_R \mathbb{G}_1$ and outputs the pair $P = (h, h^{x+my}$. Given $(g, \tilde{X}, \tilde{Y})$ and unlimited access to this oracle, no adversary can efficiently generate such a pair, with $h \neq 1_{\mathbb{G}_1}$, for a new scalar m^*, not asked to O.

2.1.2 Bilinear Pairing

$(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ is a set of cyclic groups of the same prime order p. $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ is the bilinear map, if the following three properties are satisfied:

- ▷ Bilinear: for all $g \in \mathbb{G}_1$, $\hat{g} \in \mathbb{G}_2$, and $a, b \in_R \mathbb{Z}_p$, $\hat{e}(g^a, \hat{g}^b) = \hat{e}(g, \hat{g})^{ab}$;

▷ Non-degenerate: If $g \neq 1_{\mathbb{G}_1}$, $\hat{g} \neq 1_{\mathbb{G}_2}$, then $\hat{e}(g, \hat{g}) \neq 1_{\mathbb{G}_T}$;

▷ Computable: for all $g \in \mathbb{G}_1$, $\hat{g} \in \mathbb{G}_2$, the map $\hat{e}(g, \hat{g})$ is efficiently computable.

According to the definition due to Galbraith et al. [27], If there is no efficiently computable homomorphism in either direction between $\mathbb{G}_1$ and $\mathbb{G}_1$, the bilinear map $\hat{e}$ is a type 3 pairing. If $\mathbb{G}_1 = \mathbb{G}_2$, the bilinear map $\hat{e}$ is a type-1 pairing. If there exists an efficiently computable homomorphism $\pi : \mathbb{G}_2 \to \mathbb{G}_1$, but there is no efficient homomorphism in the other direction, the bilinear map $\hat{e}$ is a type-2 pairing is that $\mathbb{G}_1 \neq \mathbb{G}_2$.

2.1.3 Pseudo Random Function

Given a seed and an argument, a secure pseudo random function (PRF) can return a string that is indistinguishable from a string generated from a truly random function. Dodis and Yampolskiy [28] proposed a secure pseudo random function that is secure under the q-DDHI assumption. The construction of the pseudo random function is as follows.

Let G be a multiplicative cyclic group of a prime order p. Let g be a generator of G. With a seed $s \in Z_p$, the Dodis-Yampolskiy pseudo-random function f is defined by $f_{g,s}(x) = g^{\frac{1}{s+x+1}}$.

2.1.4 Pedersen Commitment

Let G be a multiplicative cyclic group of a prime order p. Let g and h be two independent generators of G. A Pedersen commitment scheme [29] works as follows.

To commit an element $m \in G$, a prover chooses a random element $r \in_R Z_p$, and sends $com(m) = g^m h^r$ to the verifier. The commitment has two properties: perfect-hiding and computation-binding. The perfect-hiding property guarantees that given $com(m)$, it is feasible for a verifier to obtain any information about x. The computation-hiding property guarantees that given $com(m)$, it is feasible for a verifier to find two different pairs (m, r) and (m', r') such that $com(m) = g^m h^r = g^{m'} h^{r'}$ unless the verifier knows $\log_g h$.

The commitment scheme also has additively homomorphic property, which means given two commitments $com(m_1)$ and $com(m_2)$, a new commitment $com(m_1 + m_2)$ can be obtained by computing $com(m_1) \cdot com(m_2)$, i.e., $com(m_1 + m_2) = com(m_1) \cdot com(m_2)$.

2.1.5 Proxy Re Signature

A proxy re-signature scheme allows a semi-trusted proxy to translate one party's signature to another party's signature on the same message. It usually consists of five algorithms as follows [25] and its security can be proven under the 1-FlexDH assumption.

- ▷ Key Generation: Let $\mathbb{G}$ and $\mathbb{G}_T$ be bilinear groups of prime order p. Let $\hat{e}$ be a bilinear map $\mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$. Let g be a generator of $\mathbb{G}$. Let $H : \{0,1\}^* \to \mathbb{G}$ be a hash function. Entity i's public key is $X_i = g^{x_i}$, where $x_i \in_R Z_p$ is a random element. Entity j's public key is $X_j = g^{x_j}$, where $x_j \in_R Z_p$ is a random element.

- ▷ Signature Generation-I: Given a message m and the entity i's private key x_i , a signature of the entity i can be calculated as $\sigma_i = H(m)^{x_i}$.

- ▷ Signature Generation-II: Given a signature σ_i, a re-signing key $R_{ij} = X_i^{1/x_j} = g^{x_i/x_j}$, the entity i's public key X_i, a new signature of the entity j can be calculated as $\sigma_j = (\sigma_{j,1}, \sigma_{j,2}, \sigma_{j,3}) = (\sigma_i^t, X_i^t, R_{ij}^t)$, where $t \in_R Z_p$ is a random element.

- ▷ Signature Verification-I: Given a message m, a signature σ_i on m, and the the entity i's public key X_i, the signature can be verified the following equation.

$$\hat{e}(\sigma_i, g) = \hat{e}(H(m), X_i).$$

- ▷ Signature Verification-II: Given a message m, a signature σ_j on m, and the public key of the entity i X_j, the signature can be verified the following equation.

$$\hat{e}(\sigma_{j,1}, g) = \hat{e}(\sigma_{j,2}, H(m)),\ \hat{e}(\sigma_{j,2}, g) = \hat{e}(X_j, \sigma_{j,3}).$$

2.1.6 BBS+ and PS Signatures

BBS+ Signature. BBS+ signature is a variant of BBS signature [23] proposed in [30], which can be utilized to sign ℓ-message vector $(m_1, \cdots, m_\ell)$. Its existential unforgeability is proven against chosen message attacks without random oracles under the q-SDH assumption.

- ▷ Key Generation: Let $g, g_1, \cdots, g_{\ell+1}$ be generators of $\mathbb{G}$ with a prime order p. An random element x is chosen from $\mathbb{Z}_p$ as the secret key, and the corresponding public key is set as $y = g^x$.

- ▷ Signature Generation: A signature on messages $(m_1, \cdots, m_\ell)$ is (A, e, s), where $A = (gg_1^{m_1} \cdots g_\ell^{m_\ell} g_{\ell+1}^{s})^{\frac{1}{x+e}}$ and (e, s) are random values chosen from $\mathbb{Z}_p$.
- ▷ Signature Verification The signature (A, e, s) can be checked by the following equation: $\hat{e}(gg_1^{m_1} \cdots g_\ell^{m_\ell} g_{\ell+1}^{s}, g) \stackrel{?}{=} \hat{e}(A, yg^e)$.

PS Signature. The PS signature is a public-key signature scheme proposed in [26] and its existential unforgeability is proven against chosen message attacks without random oracles under the PS assumption [26]

- ▷ Key Generation: Let $\hat{g}$ be a generator of $\mathbb{G}_2$. $(y, x_1, \cdots, x_r) \in_R \mathbb{Z}_p^{r+1}$ is the secret key of the signer and $(\widehat{Y}, \widehat{X}_1, \cdots, \widehat{X}_r) \leftarrow (\hat{g}^y, \hat{g}^{x_1}, \cdots, \hat{g}^{x_r})$ is the public key.
- ▷ Signature Generation: A digital signature on multi-block messages $(m_1, \cdots, m_r) \in \mathbb{Z}_p^r$ is $\phi = (\phi_1, \phi_2) = (h, h^{y+\sum_{j=1}^{r} x_j m_j})$, where h is a random value chosen from $\mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$.
- ▷ Signature Verification: The signature ϕ can be publicly verified as $\phi_1 \neq \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and $\hat{e}(\phi_1, \widehat{Y} \prod_{j=1}^{r} \widehat{X}_j^{m_j}) = \hat{e}(\phi_2, \hat{g})$.

2.1.7 Dynamic Cryptographic Accumulator

The dynamic universal crytographic accumulator [31] allows a set of values to be dynamically accumulated into one value and a prover can use a witness to prove to the verifier that he/she knows a value that is accumulated not accumulated. Generally, there are two types of accumulator designs, one is designed based on the RSA assumption [32], and another is designed based on the q-SDH assumption [33].

The accumulator designed based bilinear pairing mainly consists of four parts: setup, accumulator generation, membership witness generation, non-membership witness generation, and accumulator checking.

- ▷ Setup: Let $\tilde{e} : \tilde{G}_1 \times \tilde{G}_1 \rightarrow \tilde{G}_T$ be a symmetric bilinear pairing such that $|\tilde{G}_1| = |\tilde{G}_T| = \tilde{p}$, where $\tilde{p}$ is a $\tilde{l}$-bit prime. Let $\tilde{g}$ be a generator of $\tilde{G}_1$ and $\tilde{\tau}$ is randomly picked from $Z_{\tilde{p}}^*$. Let $\tilde{G}_{\tilde{q}} \subset Z_{\tilde{p}}^*$ be a cyclic group of prime order $\tilde{q}$ satisfying $\tilde{p} = 2\tilde{q} + 1$. The public information is $\{\tilde{e}, \tilde{G}_1, \tilde{G}_T, \tilde{p}, \tilde{g}, \tilde{g}^{\tau}, \tilde{g}^{\tau^2}, ..., \tilde{g}^{\tau^n}\}$, where n is the upper bound of the accumulator.

▷ Accumulator Generation: The domain of accumulated elements is $\tilde{G}_{\tilde{q}}/\{-\tilde{\tau}\}$. To accumulate a element $\tilde{y} \in \tilde{G}_{\tilde{q}}/\{-\tilde{\tau}\}$ in the accumulator, we have $\tilde{c} = \tilde{g}_0^{\tilde{y}+\tilde{\tau}}$. Adding a value $\tilde{y}$ to the accumulator $\tilde{c}$ can be computed as $\tilde{c} = \tilde{c}^{\tilde{y}+\tilde{\tau}}$. Deleting a value $\tilde{y}$ from the accumulator $\tilde{c}$ is computed as $\tilde{c} = \tilde{c}^{\frac{1}{\tilde{y}+\tilde{\tau}}}$.

▷ Witness Generation: Assuming that the current set of elements is $\mathsf{SList} = (\tilde{y}_1, \tilde{y}_2, ..., \tilde{y}_k)$, the membership witness for $\tilde{y}_j$ can be calculated as $\tilde{wit} = [\tilde{g}^{\prod_{i=1}^{k}(\tilde{y}_i+\tilde{\tau})}]^{\frac{1}{\tilde{y}_j+\tilde{\tau}}}$. Without knowing τ, the witness can be calculated by the following steps: 1) setting a polynomial $w(x) = \prod_{i=1,i\neq j}^{k}(\tilde{y}_i + x)$; 2) expanding $w(x)$ as $w(x) = \prod_{i=0}^{k-1} c_i \cdot x^i$; and 3) calculating the witness $\tilde{wit} = \prod_{i=0}^{k-1}(\tilde{g}^{\tau^i})^{\alpha_i}$.

For another element $\tilde{y}$, the non-membership witness $\tilde{wit} = (\tilde{a}, \tilde{d})$ can be calculated as $\tilde{d} = \prod_{i=1}^{k}(\tilde{y}_i + \tilde{\tau}) \bmod (\tilde{y} + \tilde{\tau})$ and $\tilde{a} = [\tilde{g}_0^{\frac{\prod_{i=1}^{k}(\tilde{y}_i+\tilde{\tau})-\tilde{d}}{\tilde{y}+\tilde{\tau}}}]$ with knowing auxiliary information τ. Without knowing τ, the witness can be calculated by the following steps: 1) setting a polynomial $w(x)$ satisfying $\prod_{i=1,i\neq j}^{k}(\tilde{y}_i + x) = w(x)(x + \tilde{y}) + d$, where d is a constant; 2) expanding $w(x)$ as $w(x) = \prod_{i=0}^{k-1} \alpha_i \cdot x^i$; and 3) calculating the witness $\tilde{wit} = \prod_{i=0}^{k-1}(\tilde{g}^{\tau^i})^{\alpha_i}$.

▷ Accumulator Checking: An element $\tilde{y}_j$ exists in the accumulator $\tilde{c}$ can be checked by the following equation based on the membership witness $\tilde{wit}$: $\tilde{e}(\tilde{wit}, \tilde{g}^{\tilde{y}_j}\tilde{g}^{\tau}) = \tilde{e}(\tilde{c}, \tilde{g})$. An element $\tilde{y}$ does not exist in the accumulator $\tilde{c}$ can be checked by the following equation based on the non-membership witness $wit = (\tilde{a}, \tilde{d})$: $\tilde{e}(\tilde{a}, \tilde{g}^{\tilde{y}_j}\tilde{g}^{\tau})\tilde{e}(\tilde{g}, \tilde{g})^{\tilde{d}} = \tilde{e}(\tilde{c}, \tilde{g})$.

2.1.8 Zero-Knowledge Proof

The zero-knowledge proof of knowledge [34] 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. Generally, a zero knowledge proof of knowledge can be expressed in a particular notation as ZkPoK$\{(a, B) : A = g^a \wedge C = e(A, B)\}$. It denotes "zero-knowledge proof of knowledge of a and B such that $A = g^a$ and $C = e(A, B)$ hold". The proof can also be transfered to an non-interactive zero-knowledge proof based on the Fiat-Shamir heuristic [35], which has been proven secure in random oracle model. We denote the non-interactive proof as NIZK$\{(a) : A = g^a\}$. There also exist another counterpart named non-interactive signature proof of knowledge on a message M, and we denote it as NIZK$\{(a) : A = g^a\}(M)$, i.e., a signature on the message M signed by a private/public key pair (a, g^a).

Plenty of ZKPoK protocols have been proposed, in which Σ-protocols are a special format of interactive three-move ZKPoK protocols, which has the following three properties between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$:

- ▷ *(Completeness)* if $\mathcal{P}$ and $\mathcal{V}$ follow the zero-knowledge proof 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.
- ▷ *(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 $(x, w) \in R$.
- ▷ *(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.

2.2 Related Work

We first review the literature about anonymity-based privacy-preserving mechanisms, which can be applied in V2X services to protect user privacy, and then comprehensively discuss the related works about privacy-preserving schemes for smart parking services, car/ride sharing services, and crowdsourcing-based road condition monitoring services.

2.2.1 Anonymity-based Privacy-Preserving Mechanisms

Before reviewing the traditional privacy-preserving mechanisms for V2X services, we first present the identity privacy metrics for measuring the user privacy. Two most popular metrics are illustrated and we refer to [36] for more details.

Anonymity Set Size. One user's identity is indistinguishable from a set of users' identities, and this set is called anonymity set (AS). Anonymity set size means the size of the anonymity set, i.e., the number of users that this set has. Commonly, the size of the anonymity set represents the level of the identity privacy [37] and it is simple and easy to calculate the anonymity set size to measure the privacy level. However, the prior knowledge of adversary is not considered in this metric.

Entropy of Anonymity Set Size. Based on the AS size, a more detailed privacy definition has been proposed with the help of entropy. The entropy of the anonymity set, compared to the AS set, is more accurate and allows expressing the adversary's knowledge about each user of the anonymity set. The following equation formally defines the entropy of anonymity H_p [38].

$$H_p = -\sum_{N}^{i=1} p_i log_2 p_i$$

where p_i refers to the probability of user i being the victim. If all N users have a same probability to be attacked, i.e., probabilities are uniformly distributed over the anonymity set, the entropy then achieves its maximum as $H_{max} = log_2 N$, but in reality, the inference probability p_i is different to be determined.

Many anonymous schemes have been proposed for achieving privacy preservation for V2X services, which can mainly categorized into two classes: one is designed based on pseudonym management and traditional public key infrastructure (PKI), and another is designed based on group signature. We review some relative works in the field as follows.

PKI-based Mechanisms with Pseudonym Management. The anonymity of users can be achieved with the conventional PKI architecture. The users can be assigned a large number of X.509 public key certificates as pseudonyms and the corresponding private keys. These certificates are the unlinkable pseudonyms of users and can be generated by a trusted certificate manager. When users communicate with other users or entities in V2X-based applications, they can sign the message with one of the private key and attach the created signature, as well as the corresponding certificate, to the message. Receivers can verify the received certificate using the root certificate of the certificate manager and verify the signature using the received certificate to validate the sender without knowing the sender's real identity. The first idea of PKI-based anonymous mechanisms comes from [39], and then many following researches have been proposed. Among them, how to change the pseudonyms of users is one of the hot topics and the pseudonym change rate has a great impact on the communication, computation, storage overhead, and the level of privacy. Wiedersheim et al. [40] showed that the simplest pseudonym change, that a user changes its pseudonym according to a fixed and periodic schedule, is not enough with regard to anonymity. Hence, different kinds of pseudonym change polices have been proposed. Eckhoff et al. [41, 42] proposed a new pseudonym management scheme that every user maintains a set of pseudonyms of static size, each pseudonym can be used for the specific time slot and these pseudonyms can be exchanged between users. In their schemes, the change rate is still fixed and the prediction of change rate will easily reveal the relationship between pseudonyms. To solve this issue, Pan et al. [43]

proposed a pseudonym change scheme where the users randomly change their pseudonyms. As a result, the next pseudonym change is not easy for prediction. However, anonymity is still not guaranteed if only one or few users change pseudonyms at a specific time, because the nearby users who keep the same identity will leak the pseudonym change information. Consequently, the idea of silent random period has been proposed [44, 45] where users remain silent for a short period after they change their pseudonyms, although it has no explicit definition. The silent-period-based schemes are more complex compared to the previous schemes. More questions come up, such as how long for the silent period and where to change the pseudonym to satisfy the silent requirement. A good silent period strategy should balance a trade-off between privacy and safety. Therefore, Lu et al. [46, 47] analyzed the social spot based pseudonym change and proposed a social spot based pseudonym change strategy to maximize the privacy in terms of anonymity size. Furthermore, Yu et al. [48] proposed a scheme called MixGroup, where users constructs extended pseudonym-changing regions from the social spots and are allowed to successively exchange their pseudonyms. From another point of view, Emara et al. [49] proposed a context adaptive pseudonym changing scheme which allows a user to decide autonomously when to change its pseudonym and how long it should remain silent to ensure unlinkability. There also exist other pseudonym change schemes based on different characteristics, e.g., the users's reputation [50]. Despite different strategies proposed for pseudonym changing, it still remains unclear which strategy is the most effective in practice.

Group-signature-based Anonymous Schemes. To mitigate the overhead of managing a mass of certificates for users and updating the pseudonyms, Calandriello et al. [51] proposed to use a group-signature-based method to enable each user to generate and certify their own pseudonyms without interacting with the certificate authority. Basically, their scheme allows a authority to distribute traditional public key certificates using group signature. More following works [52, 53, 54] go further step and design more comprehensive anonymous authentication schemes for V2X services. For example, Lin et al. [52] proposed the GSIS scheme which first presents the anonymous authentication protocol for V2X communications. Their scheme brings up a better way to meet the anonymity and traceability requirements rather than storing all the certificates in the vehicle, in contrast to PKI-based schemes.

Group signature [55] enables any group member in a specific group to produce a signature on behalf of the group, i.e., the signature can be verified with a group-oriented public key. The group-signature-based scheme provides user privacy protection as signers are anonymous within the group, which can perfect match the privacy requirement of users when communicating with other as well as passing the authentication. At the beginning, two messages signed by the same vehicle are not linkable as one cannot determine if two

messages came from the same or different user. Thus, the linkable group signature [56] was proposed for this goal and can achieve the linkability property. Also, the linkable group signature supporting dynamic group [57] was also proposed recently, which provides more flexibility for V2X services. However, a group-signature-based approach, when applying to V2X services, also has its disadvantage. Although the group signature approach does not require each user to store a large number of certificates, the unrevoked users have to update the revocation lists with the group manager when some users in this group are revoked. To solve the issue of revocation overhead, Lu et al. [58] proposed an ECPP scheme to deal with the issue of growing revocation lists. Their scheme allows users to obtain the short-term pseudonyms via group signature, and to use the short-term pseudonyms for communication and authentication. If a user is revoked, other users do not need to update the revocation lists since the revoked users cannot renew its short-term pseudonyms. However, the group-signature-based methods face a big issue about revocation efficiency. There are basic two approaches to support user revocation. The first approach is to achieve verifier-local revocation [59, 60]. The revocation lists are distributed to verifiers and when receiving a group signature, a verifier can traverse members in the revocation list to judge whether a user who generated the signature is revoked. The approach supports backward unlinkability assuming that there exist a trusted revocation manager who can update the revocation list according to time. Another is to achieve accumulator-based revocation [61, 62]. All revoked users are accumulated into one value, and non-revoked users can prove to the verifier through zero-knowledge proof that they are not accumulated. However, this method requires all users share a large number of public parameters which is linear to the size of revocation list in most cases. Currently, there is no solution for this issue.

2.2.2 Privacy-Preserving Schemes for V2X Services

In this section, we review privacy-preserving schemes of three V2X services, which are most relative to this thesis, including secure and privacy-preserving smart parking, secure and privacy-preserving car/ride sharing, secure and privacy-preserving road monitoring.

Secure and Privacy-Preserving Smart Parking: Many researches in V2X-based smart parking scenarios related to security and privacy have been proposed in recent years. Yan et al. [63] proposed a secure and intelligent parking service, which enables users to securely book parking slots. They also model the parking process as birth-death stochastic process and design a new business model for the parking system based on the prediction of revenues. Biswas et al. [64] proposed a secure and privacy-preserving car parking assistance application using priority-based vehicular communications. Their scheme is designed based on a modified version of Elliptic Curve Digital Signature Algorithm (ECDSA) and

offers message authentication and user anonymity for a parking service. Unlike PKI-based message authentication approaches, their scheme uses the geographical location of an entity to validate a received message. Lu et al. [65, 66] presented an intelligent secure and privacy-preserving parking scheme through vehicular communications. They use roadside units (RSUs) to localize the vehicles and assist users to find vacant parking spaces in a privacy-preserving way. That is, the users use the pseudonyms, assigned by a third-party trust authority, to protect their privacy when communicating with the RSUs. Ni et al. [67] proposed a cloud-based privacy-preserving parking navigation system in VANETs to locate accessible parking spots for users. They utilize anonymous authentication to protect the privacy of the users in VANETs and additionally construct a navigation system for users. Namely, users can retrieve the protected navigation responses from RSUs when the vehicles are passing through the RSUs. In their extended version [68], they provided more details about the navigation performance analysis and demonstrate the practicality of their scheme. Garra et al. [69] proposed a privacy-preserving pay-by-phone parking system by implementing an anonymous e-coin-based payment protocol. Their scheme can keep the payment information secret while providing the evidence that the payment has been finished without leaking the user's privacy. Through achieving anonymous payment, a parking slot can be locked and the user can enjoy the parking service afterwards. Borges et al. [70] proposed a new a privacy-preserving pay-by-phone parking system based on Garra et al.'s scheme [69], and offering the same privacy as [69]. Furthermore, the new system can tolerate that the mobile devices of users fall out of coverage while their cars are parked, which is not achieved in [69]. Ni et al. [71] proposed a secure and privacy-preserving automated valet parking protocol for self-driving vehicles. Their protocol not only achieves anonymous authentication but also supports multi-factor authentication for reducing the risks of vehicle theft and preventing the privacy leakage of users. Chatzigiannakis et al. [72] proposed a privacy-preserving smart parking system based on elliptic curve cryptography and zero-knowledge proof. They also study the performance of the system in an real-world outdoor IoT testbed to demonstrate the system's feasibility.

As blockchain has become becomes a hot platform to achieve some security properties, some secure parking systems are designed atop blockchain. For example, Hu et al. [73] proposed a blockchain-based framework for parking-management, which can preserve the privacy of its users, without relying on a reliable third-party entity. The parking information is managed in a consortium blockchain and can be shared with users. The privacy is protected based on the natural characteristic of the consortium blockchain, i.e., only authorized entities can access the parking information with permissions. Wang et al. [74] proposed an airbnb-like privacy-enhanced private parking spot sharing scheme. All parking spaces are managed by individuals but can be shared in a consortium blockchain.

Their scheme achieves decentralized anonymous credentials and confidential anonymous payment with a variant Monero to protect user privacy. Amiri et al. [75] proposed a blockchain-assisted privacy-preserving smart parking system, where users can retrieve the parking offers nearby without leaking privacy through a method named private information retrieval. Ahmed et al. [76] proposed a blockchain-based architecture for integrated smart parking systems. Several parking service providers can work together to construct a consortium blockchain to provide parking services for users. The user privacy is achieved through fine-grained access control, i.e, only vacant space information can be accessed by parking service providers.

Different from the above-mentioned schemes, in this thesis, we mainly focus on resolving the conflict between privacy and linkability and consider a special attack "Double-Reservation Attack" in the parking system. Although most of the existing secure parking schemes address the privacy concerns of users by utilizing pseudonym-based mechanisms or group-signature-based mechanisms, they somehow ignore the malicious behavior where a user could occupy all parking spaces without being detected. This attack happens in an automated valet parking scenario, where a user is only allowed to reserve one parking space for his/her automated vehicle. Some schemes offer a traceability function by introducing a trusted third party to help the parking service provider trace a user's malicious behavior, but it cannot prevent such an attack in advance and needs to rely an trusted party, which is not practical to be deployed in the real world.

Secure and Privacy-Preserving Car/Ride Sharing: Plenty of works have been proposed recently related to secure and privacy-preserving ride sharing, while few works aim to solve the security and privacy issues in car sharing services.

To address the security and privacy issues in ride sharing (carpooling or ridehailing) services, Kanza et al. [77] proposed a ride-hailing service powered by cryptocurrency and blockchain to preserve location privacy of users and guarantee pseudonymity of users and drivers. They scheme use blockchain-inherited pseudonymity to protect users' and drivers' privacy and only requires users to report coarse areas to hide the exact locations to protect location privacy. Ni et al. [78] proposed an anonymous mutual authentication scheme for carpooling systems. Their scheme uses BBS+ signature to achieve anonymous mutual authentication between user and driver and relies on a trusted judger to achieve traceability. Hallgren et al. [79] proposed a privacy-preserving scheme to achieve user-driver matching through private set intersection. Zhao et al. [80] studied privacy leakage for drivers in a ride sharing system by measuring and analyze privacy attacks in 20 ride-hailing apps, , e.g., tracking the driver's daily routine, uncovering employer status and preference, and business information leakage. Khazbak et al. [81] proposed a privacy-preserving scheme to protect user privacy when matching his/her locations with drivers. They use drivers' location to

generate the spatial cloaking to match the user's locations and also design a probabilistic driver selection algorithm to reduce the background knowledge of attackers. Pham et al. [13] a privacy-preserving yet accountable ride-hailing service. Their scheme depends on a somewhat homomorphic encryption mechanism to encrypt the locations of users and drivers, and user-driver matching can be achieve through the computations on encrypted data. Goel et al. [82] proposed a ride-hailing service that improves the safety while preserving user privacy. Their scheme helps in choosing the pick up/drop off locations for passengers from a fixed set and use Voronoi diagram-based k-anonymity model to preserve user privacy. They also proposed a privacy-aware dynamic ride sharing scheme [83] by obfuscating on the user's and the driver's locations and time. Luo et al. [84] proposed a privacy-preserving ride-matching over road networks for online ride hailing service. They improve the accuracy of user/rider matching while preserving their privacy by introducing the road network and designing the matching algorithm based on homomorphic encryption and garble circuit.

To address the security and privacy issues in car sharing services, Symeonidis et al. [85] 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 [86], 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) [87], 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. [88] 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.

Different from the above-mentioned schemes, in this thesis, we mainly focus on resolving the conflict between privacy and accountability in a car sharing service. These existing schemes address the security and privacy issues of a car sharing service in terms of different aspects such as digital forensics and authentication security, but none of them considers the accountability issue and just assume that there exist a trusted entity to assist the

car sharing service provider in achieving accountability and tracing the malicious behavior of customers. To get rid of the deployment of a trusted party, some other works deploy decentralized authorities to achieve accountability in a more secure and robust manner. As long as the majority of the authorities are honest, the privacy and accountability can be guaranteed at the same time. However, they do not apply this mechanism in the car sharing scenario, and they do not consider a strong attack model, where eventually an adversary could compromise a majority of the decentralized authorities if they do not change.

Secure and Privacy-Preserving Crowdsourcing-based Road Condition Monitoring: To achieve privacy preservation in a scenario of crowdsourcing-based road condition monitoring, there exist two major approaches: 1) anonymizing each user's road condition report such that a monitoring service provider cannot distinguish the source of the report; and 2) encrypting a road condition report such that only authorized authorities can decrypt and obtain the report. In short, the approach 1) is to hide the user identity (anonymization) while the approach 2) is to hide the report content (encryption). For example, Li et al.[89] proposed a privacy-preserving traffic monitoring scheme via fog-assisted vehicular crowdsourcing, where users who make the report are anonymized but can still be authenticated by a monitoring service provider. They mainly utilize a group-signature-based method to achieve anonymous authentication between users and the monitoring service provider. Instead of utilizing group signature, Basudan et al.[90] proposed a privacy-preserving vehicular crowdsensing-based road surface condition monitoring scheme, where users utilize registered pseudo-identities to authenticate themselves to a monitoring service provider and then make reports. They are two representative works that achieve privacy preservation based on the anonymization.

Different from them, Wang et al. [91] proposed a privacy-preserving cloud-based road condition monitoring scheme, where users encrypt their reports before submitting them to a monitoring service provider and only a root authorized entity can decrypt them with the assistance of the monitoring service provider. They mainly utilize a public key encryption primitive with equality test in their scheme, which enables users to encrypt their reports and the monitoring service provider to statistically aggregate the reports without decryption. Similar to Wang et al. [91], there also exist other schemes such as [92, 93, 94, 95], and these schemes' design concept is to conceal the user's report to achieve privacy-preservation based on encryption. However, the above-mentioned schemes, although achieving anonymization or pseudonymization to some extent, do not consider data manipulation attacks or apply reputation management to filter a faked or biased report based on the user's reputation score to improve the data availability.

Therefore, on the premise of anonymity, we discuss two major reputation management approaches that are widely adopted in crowdsourcing-based road condition monitoring

scenarios. The first approach is to rely on one centralized third party or decentralized third parties to manage the reputation scores for anonymous users. Zhu et al. [96] proposed a privacy-preserving trusted-based traffic monitoring architecture, where users are anonymized during the data reporting process and a trusted authority is introduced to maintain the reputation scores of the users. This method suffers from the single point of failure and requires the trusted authority to be always online, which is not practical. Similar to Zhu et al. [96], Li et al. [97] and Yu et al. [98] proposed pseudonym-based crowdsourcing schemes where users are pseudonymized but linkable. In their schemes, the reputation scores are bound with the pseudonyms and are published on a decentralized blockchain. Differently, Wu et al. [99] proposed a trustworthy and privacy-aware mobile crowdsensing scheme that enables an honest-but-curious group manager to manage the reputation scores for users. The users can authenticate themselves to a monitoring service provider and can update their reputation scores by communicating with the group manager. Through a blind signature method, even though the monitoring service provider and the group manager collude together, the anonymity property still holds for users. However, their scheme reveals the mapping between reputation scores and users if the group manager is curious. this information can be further utilized by a curious service provider to track a user's reputation change although the reputation score is blurred. From another perspective, Zhao et al. [100] proposed a tracking-resistant anonymous reputation management scheme that can be applied in the crowdsourcing-based road condition monitoring scenario, where multiple anonymity providers are introduced to maintain the reputation scores for users. As long as one anonymity provider is honest, the mapping between reputation scores and users is hidden. Their scheme has stronger privacy guarantee but is more time-consuming as each anonymity provider needs to shuffle users' reputation scores and prove that their behavior is correct.

The second approach does not rely on third parties and the anonymous users can self-maintain their reputation scores locally. Yi et al. [101] proposed a privacy-preserving mobile crowdsourcing scheme with anonymous reputation management, which is designed based on a blind-signature-based method. Their scheme achieves privacy preservation, and they are designed based on RSA assumption. Nevertheless, they do not consider collusion attacks among users, i.e., each user can share his/her anonymous reputation score with other users without being detected since the reputation scores are not bound with the anonymous identity credential. Both Ni et al.'s [102] and Hartung et al.'s [103] schemes address the collusion issue but they utilize different methods. Ni et al. [102] proposed a privacy-preserving mobile crowdsourcing scheme, which is designed based on a group-signature-based method. Hartung et al. [103] proposed a privacy-preserving credit points collection scheme, which is designed based on a structure-preserving-signature-based

method. They have sophisticated protocol designs such that users can self-manage and update their reputation scores with the help of a monitoring service provider. But these schemes cannot be straightforwardly applied in a crowdsourcing-based road condition monitoring scenario. The reason is that they assume that users will automatically and spontaneously update their reputation scores after completing some tasks, e.g., road condition reporting, but users may behave maliciously and does not choose to update their reputation scores since they know they uploaded a faked or biased report and does not want their reputation score to be downgraded. In addition, there exist other reputation management schemes such as [104, 105], their motivation is different, since they mainly address the reputation management issue for service vendors but not users, e.g., managing reputation scores for restaurants.

In this thesis, we mainly focus on resolving the conflict between privacy and reliability in a crowdsourcing-based road monitoring service. Different from some existing schemes, a trusted third party or decentralized trusted third parties are not desirable since they rely on impractical assumptions and may be compromised by adversaries to break user privacy and accountability. From another point of view, Yi et al.'s [101], Ni et al.'s [102], and Hartung et al.'s [103] schemes cannot be straightforwardly applied in the road monitoring scenario, as their schemes have some vulnerabilities and are not fit for the reputation-based solutions due to some security issues such as suffering from collusion attacks and reputation manipulation attacks, which will cause serious security concerns.

2.3 Summary

In this chapter, we have briefly reviewed the preliminaries, including number-theoretic problems, bilinear pairing, pseudo random function, Pedersen commitment, proxy re-signature, BBS+ signature, PS signature, dynamic accumulator, and zero-knowledge proof. Also, we have given a comprehensive survey on the existing works about anonymous mechanisms and privacy-preserving schemes for V2X services, including secure and privacy-preserving smart parking, secure and privacy-preserving car/ride sharing, secure and privacy-preserving crowdsourcing-based road condition monitoring. From the comprehensive literature review, we are aware that the privacy challenges have not been well-addressed. In the following chapters, we will introduce several countermeasures to address the challenging issues and reach the research objectives of this thesis.

Chapter 3

Privacy-Preserving Parking Reservation for Automated Valet Parking Services

3.1 Introduction

Parking, as one of the perennial headaches of urban life, is a common but especially vexing problem for big cities. This hassle is not only caused by the fast-growing number of vehicles, but also by the unbalanced distribution of parking lots and the lack of a parking guidance system. Hence, a fantastic solution, automated valet parking (AVP) [106] has been proposed recently, which relies on the autonomous driving techniques to avoid the defects of valet parking. Taking the AVP solution of Daimler-Benz company as an example [107], an automated valet parking mission starts when a driver drops the AV in a designated drop-off area, and then he/she can monitor and control the autonomous vehicle (AV) via the smartphone until the parking task is accomplished. On one hand, the sensors installed in the parking lot can help steer the parking process; on the other hand, the AV itself can perform safe driving manoeuvres in response to the commands from the parking infrastructure and stop the vehicle if an emergency situation takes place.

Though the Daimler-Benz's AVP system has been licensed by the government, it is still an incomplete autonomous parking solution. It just achieves the "partial self-parking functionality" since the AV has to be dropped at a drop-off area but not anywhere else. Similarly, another automotive company, ZongMu Technology [108] has just released its self-parking products, and announced that its goal is to achieve a remote automated valet

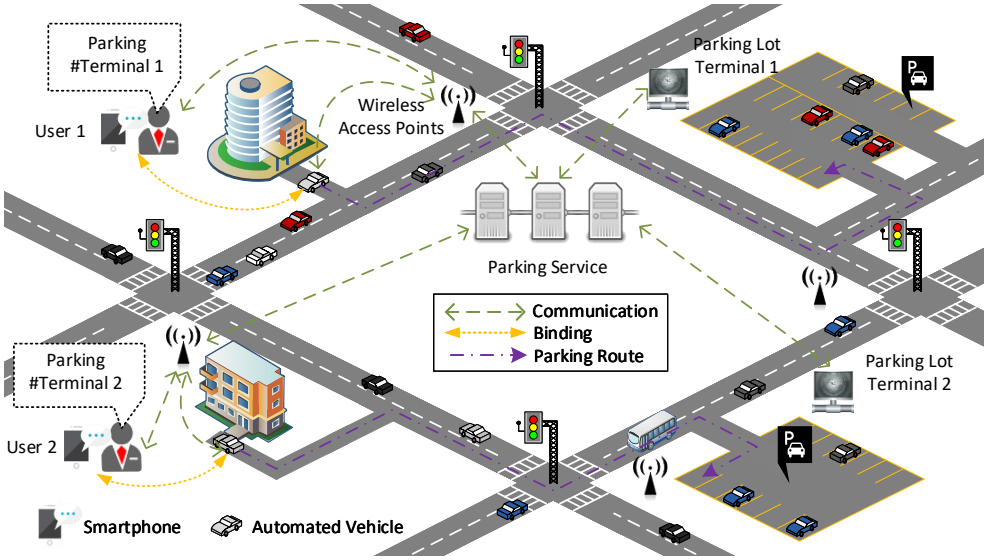


Figure 3.1: A high-level remote automated valet parking scenario

parking solution step by step using the close-to-market sensors. As shown in Figure 3.1, when a driver has reached his/her destination (e.g., place of work, gym, or hospital), he/she can leave the vehicle and control the self-parking process by the smartphone remotely, e.g., following the parking route in a high-level parking scenario. Considering the low velocity of AV (up to 30 km/h) and the light traffic situation, the deployment of AVP is mostly limited to the immediate vicinity of the location where the driver leaves the vehicle, which will reduce the requirements regarding the capabilities of AV significantly.

Generally, an AVP system can be virtualized as three subsystems [109]: mapping, perception, and communication as shown in Figure 3.2. The mapping subsystem involves the localization module, the planner module, and the map module: the localization module supports GPS and GPS-denied localization to avoid collisions and plans appropriate motions; the planner module is responsible for generating an optimal trajectory from a start position to a destination, including on-road trajectory and the trajectory into the parking bay inside the parking lot; the map module creates a high-precision 3D geometric map which contains the detailed on-road and parking lot information. The perception subsystem consists of the sensing module and environment modeling module: the sensing module collects the sensing information from the LIDAR, radar and multiple cameras; the environment modeling module constructs a dynamic environment model based on the sensing information, such as detecting and tracking moving vehicles and pedestrians. The communication subsystem takes charge of sending/receiving the messages/commands to/from

the parking service provider, the parking lot terminal and the driver's smartphone. The above modules are hot research topics for an AVP system, but less works have been done in the related area of security and privacy issues.

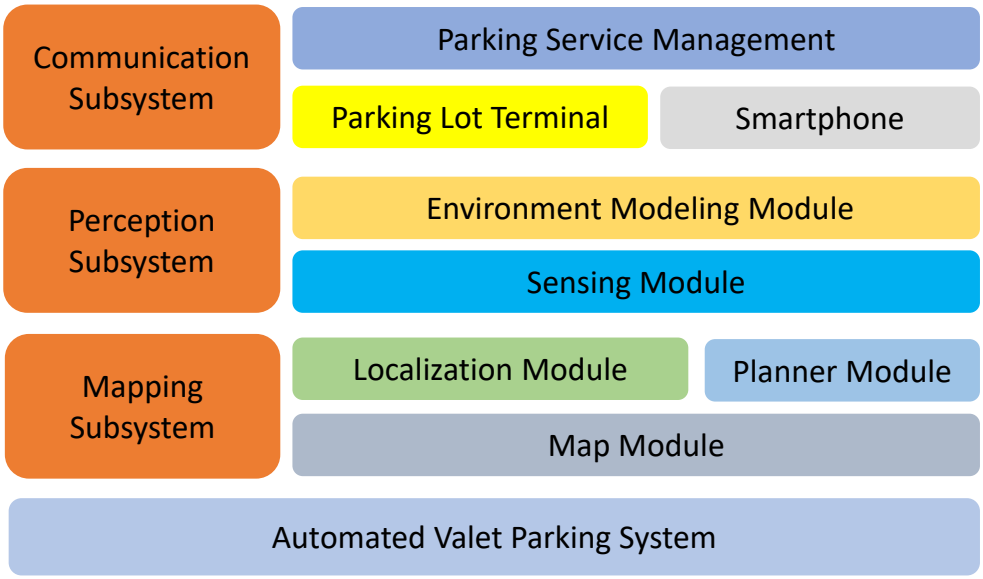


Figure 3.2: The diagram of an AVP system

Different from the traditional parking systems, the AVP system requires the driver to reserve a nearby vacant parking space in advance so that his/her vehicle can park itself autonomously without human intervention. However, this reservation procedure is under risk of privacy violation. In specific, when parking is required, the AVP system requires the AV to report its current location to the parking service provider (PSP) via the communication subsystem so that a better parking choice can be offered to locate an optimal nearby parking space for that vehicle. In this situation, the PSP will learn the personal and location-privacy-sensitive information, such as the most visited places of the vehicle, by investigating its uploaded locations [110, 111], which means that the driver's location privacy has been compromised. To address the privacy issue, a naive way is to introduce the anonymous mechanism into the AVP system: each autonomous vehicle will have plenty of pseudonyms which can also be authenticated by the PSP to protect the driver's privacy. Since the location privacy attacking method [112] needs at least four continuous location points in a trace, with both spatial relation and temporal relation, to identify a particular driver, the anonymous mechanism is effective due to the discrete characteristic of the parking behavior. In the parking scenario, the PSP cannot obtain

four continuous location points from the AVP system because the average time interval between two parking demands is long enough.

From another perspective, the reserved parking space will be kept until the automated vehicle finishes the parking process or the reservation is expired, which gives the chance for malicious drivers to launch the "Double-Reservation Attack". The drivers cannot be assumed to behave honestly in order that he/she can launch an attack with the aid of the anonymous mechanism. Namely, the driver, as an adversary, would like to maximize his/her interest when making the parking space reservation. Despite the fact that the vehicle only needs a parking space, it could pretend to be many vehicles and preoccupy all possible parking space in the nearby parking lots. This attack could also be launched by competitors that run the similar parking services. When all available parking slots of a parking service are occupied, users may choose to use its competitors's parking services and the competitors could attract more users to gain more benefits. Under such condition, it is very difficult for the PSP to detect and track the attack due to the anonymity if no trusted third party exists.

In this chapter, to address the above-mentioned challenges in the parking reservation scenario, we propose a novel privacy-preserving reservation scheme for securing AVP system, which can protect the users' privacy using cryptographic techniques and prevent the "Double-Reservation Attack" in a simple but efficient way. The fundamental intuition of our scheme is to design a mechanism which makes sure that each anonymous user must have only one valid reservation token at any moment, and the token can only be used for booking one vacant parking space once. The contributions of this chapter are summarized as twofolds.

- ▷ We define the system and security model for a reservation/parking case of an AVP system without a trusted third party. Following the models, we propose a privacy-preserving parking reservation scheme based on four building blocks: zero-knowledge proofs of knowledge, geo-indistinguishable mechanism, proxy re-signature, and bloom-filter data structure. The proposed scheme does not only protect the driver's identity privacy and location privacy, but also prevents the "Double-Reservation Attack".
- ▷ We run extensive simulation to evaluate our scheme's performance in terms of computational costs, communication overheads and storage costs, by comparing our scheme with three previous proposed schemes [113, 114, 115]. The comparison results show that our scheme is more efficient. Additionally, we establish a WiFi-based testbed and run some experiments to further study our scheme's performance in the real-world environment, demonstrating its practicality.

3.2 Models and Design Goals

In this section, we define the system model, security model, and also identify the design goal for a reservation/parking case of an AVP system.

3.2.1 System Model

Our system model mainly consists of the following four entities: the parking lot terminal (PLT), the parking service provider (PSP), the autonomous vehicle (AV), and the smartphone (SM) as shown in Figure 3.3.

- ▷ **Autonomous Vehicle** (AV): the AV is a critical and mobile component for an AVP system. With the support of self-driving techniques, smart vehicles can achieve automated parking operations. The AV is supposed to have an autonomous capability (can be low-level to high-level depending on different situations) in automated driving and parking modes, and also has a communication ability based on cellular network (e.g. LTE V2X [116]) so that it can be directly connected with other entities in the network. The AV is owned by and under the control of a driver (a.k.a user), and the user could command the AV to accomplish some tasks, such as self-parking.

- ▷ **Smartphone** (SM): the SM is an intelligent portable device, which has a restricted computational capability and is bound with the AV. Obviously, any well-designed smartphone is able to communicate with others through the internet (e.g., WiFi). The SM is owned by and under the control of a driver (a.k.a user), and the user could install the parking application and use this application to complete the reservation process.

- ▷ **Parking Service Provider** (PSP): the PSP is a bunch of online servers who provide the on-demand parking service for the users, involving finding nearby parking space, making parking space reservation and other superior services. These services, offered by a parking management company, are the subscription services. Only the registered user who pays for the membership fee can enjoy these convenient services. Furthermore, the services could be published to the users as a smartphone application, like an Android/IOS App.

- ▷ **Parking Lot Terminal** (PLT): the PLT is a terminal deployed by the owner of the parking lot, which is responsible for monitoring and managing the parking lot through IoT devices (e.g. cameras and sensors), such as recording the parking space

status and charging the fee for the parking car. In addition, the PLT will upload its parking lot's real-time status (e.g. the parking fee, the unoccupied parking space and the high-definition map) to the PSP so as to attract more vehicles. Meanwhile, the PSP could utilize this information for the parking lot recommendation.

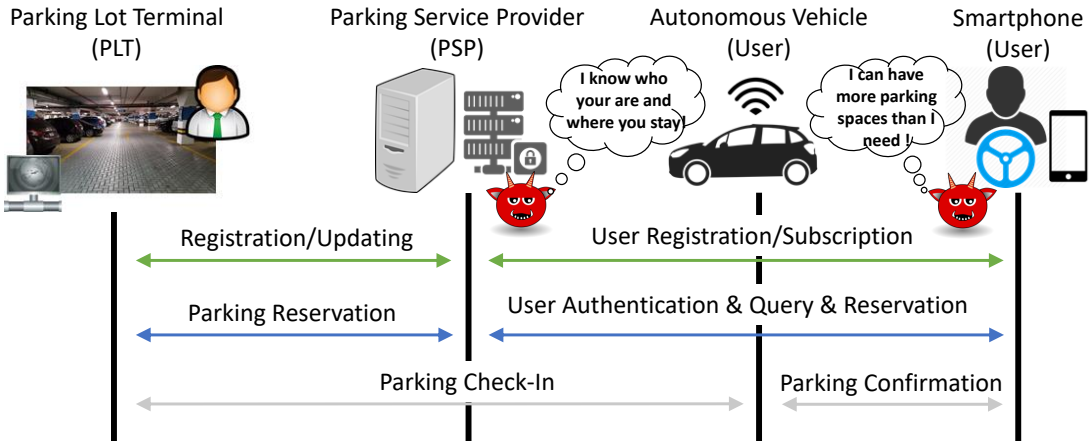


Figure 3.3: A system model of reservation and parking case for AVP

To clearly illustrate a reservation and parking case, only one type (reservation then parking) of AVP parking services is discussed detailedly in this chapter as shown in Figure 3.3. Above all, the users should download the parking App in their SMs and register themselves at the PSP. Moreover, the valid parking lots' public information is collected by the PSP in real time. When a user intends to find a parking space, he/she first needs to pass the authentication as a registered subscriber using the installed App, and then queries based on his/her current location and makes a parking space reservation according to his/her requirements. Finally, the user will let his/her AV check in and park at the reserved space by communicating with the AV through the SM, and gets the confirmation when the parking process is over. We omit the picking-up process for a parking service since it is beyond the scope of this chapter.

3.2.2 Security Model

The PSP is *honest-but-curious*, i.e., it follows the protocols, but is also curious about the user's privacy by launching passive attacks. We give an explicit definition of the user's privacy for the autonomous valet parking service at the intuitive level. Specifically, we

desire our privacy-preserving reservation scheme to have the following two properties to protect the user's identity privacy:

- ▷ **Pseudonymity**: the PSP will not be able to identify the unique user's real identity that generates a particular reservation/parking request/query. The only exception is at the stage of registration, and the users have to reveal their real identities to the PSP to prove themselves as the valid users.
- ▷ **Unlinkability**: the PSP cannot correlate a user's any two reservation/parking sessions. With the knowledge of two sessions' authenticated credentials, two sessions cannot be linked any better than guessing even if the they come from the same user.

Pseudonymity and unlinkability could be summarized as anonymity to some extent, which is a simple but an effective way to protect the user's identity privacy. To further enhance the user's location privacy, the property named geo-indistinguishability [117], is also utilized to protect from the location-based statistical analysis attack in our system.

- ▷ **Geo-indistinguishability**: The location obfuscation mechanism used by the users satisfies ε-geo-indistinguishability.

From another point of view, the users should not be totally trusted because they are selfish to launch the attack driven by self-interest and gain the benefits. In our security model, the selfish users may deliberately reserve/occupy many parking spaces at once since they are anonymous and cannot be tracked, although they merely need one parking space. Therefore, we introduce this new primitive named **"Double-Reservation Attack"** in the reservation process for an AVP system.

In addition, we assume that the PLT does not collude with the PSP to compromise the user's privacy. Since this kind of collusion attack has become a physical attack, and it cannot be entirely solved based on secure protocols. Supposing that the PLT, colluding with the PSP, can use the cameras to record a user's parking AV, it would definitely approve the real identity (car's exclusive license number) of a user to the PSP, no matter what protocols are proposed to protect the user's privacy. In this situation, not only should the secure protocols be designed but also the privacy law should be applied to forbid the privacy violation behaviors of the parking company in the physical world, which is out of scope of this chapter.

However, there exist two main limitations in our security model: 1) the exact probability that two pseudonyms of a user can be linked depends on various "side-information". The

linking probability does not just rely on the anonymity but also the user's requirements and behaviors. These "side-information" could be linked to identify the unique human [118]. Nevertheless, note that the common parking issues always happen in the most populous regions (a lot of vehicles needs to be parked nearby and cannot easily find a parking space) and in a discrete way (a driver usually will not have two continuous reservation/parking requests), so there could be plenty of similar parking requests during a short period at the adjacent locations, which will help relieve this limitation; 2) there might be other ways, outside our security model, where a user's privacy can be violated. For example, the original IP address in the cellular network could be a single tag to identify the user (a.k.a, network traffic analysis). To cope with the issue, our scheme could be coupled with other techniques (e.g., the anonymous network, Tor [119]) to guarantee the user's privacy.

3.2.3 Design Goals

Under the aforementioned system model and security model, our design goal is to propose a privacy-preserving reservation scheme for autonomous valet parking. In particular, the following three objectives should be achieved:

- ▷ **Security**: the security requirements mentioned above should be satisfied. Namely, not only is the user's privacy protected, but also the reservation system must only allow the user to book one parking space at one time, to prevent the "Double-Reservation Attack".
- ▷ **Functionality**: the basic functions supporting reservation for an AVP system should be achieved. The basic functions covers user subscription, user authentication and parking reservation/cancelling, etc.
- ▷ **Efficiency**: the proposed scheme should be efficient. To implement the reservation scheme for a real-world AVP system, both the security and efficiency issues should be considered to locate a trade-off solution.

3.3 The Proposed Privacy-Preserving Parking Reservation Scheme

In this section, we first define the pieces of our privacy-preserving reservation scheme and then present a construction for the proposed scheme based on four basic building

blocks: zero-knowledge proofs of knowledge [120], geo-indistinguishable mechanism [117] and proxy re-signature [25], and bloomfilter data structure. For easier reading, we also give the description of notations to be used in our scheme in Table 3.1.

Table 3.1: Notations frequently used in our scheme

Notation	Definition
λ	the security parameter
$\mathbb{G}$,$\mathbb{G}_T$	two cyclic multiplicative groups
p	a large prime whose length is λ
g	a generator of $\hat{\mathbb{G}}$
$H(), H'(), \hat{H}()$	three non-cryptographic hash functions
$(a, A = g^a)$	the PSP's private key and public key
X, Y, Z	$X = g^x$, $Y = g^y$, $Z = g^z$ and $x, y, z \in Z_p$
$e(.,.)$	a non-degradable bilinear mapping
μ	the daily verification day
Ω, Ξ, Ψ	three sets for storage
(B, g^b)	the PLT's private key and public key
R_{ab}	the PLT's resignature key
$cred$	the anonymous credential of user
$Timestamp$	the current timestamp
$SESS$	the token of each parking session

3.3.1 Design Overview

There are three major pieces of the proposed scheme in an AVP system, as shown in Figure 3.4, including **System Setup**, **Service Phase**, and **Parking Phase**.

- ▷ **System Setup**: ① the PLT registers itself at the PSP, and updates its real-time parking condition for the PSP periodically; ② the user registers himself/herself at the PSP; ③ the registered user subscribes to the services based on the online payment, such as Alipay or Paypal, and acquires the anonymous subscriber credential by smartphone.
- ▷ **Service Phase**: ① the user authenticates himself/herself to the PSP as a registered subscriber via smartphone; ② the user queries and searches the nearby parking lots

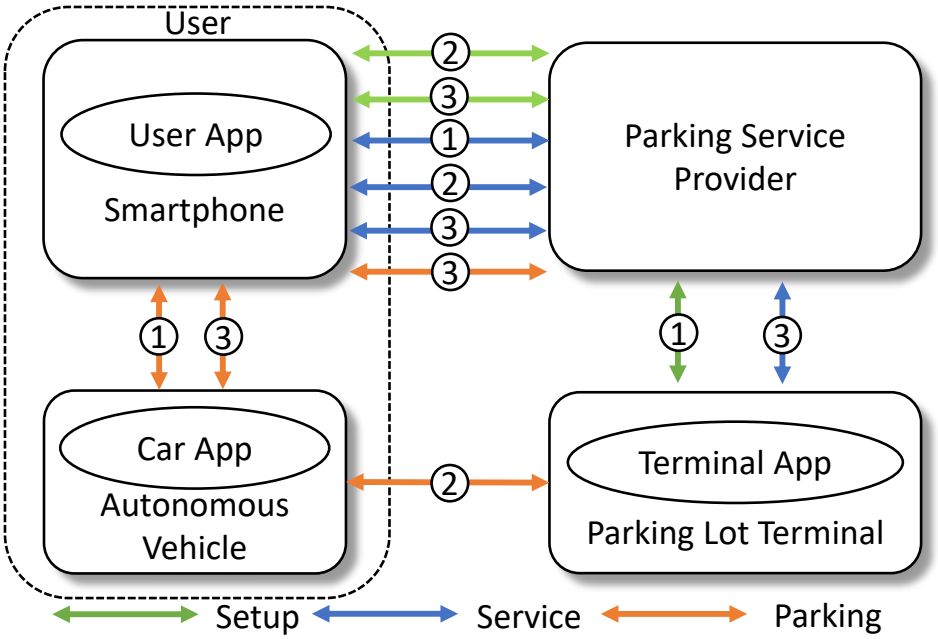


Figure 3.4: The communication framework of AVP

for the vacant parking spaces, and then choose one vacant parking space according to the requirements; ③ the user sends the reservation request to the PSP and the PSP makes the parking reservation at the PLT, and then the parking permit generated by the PLT is sent back to the user.

▷ **Parking Phase**: ① the user forwards the permit to the AV by smartphone and commands the AV to park at the reserved parking space in an autonomous driving model; ② the AV checks into the parking lot based on the permit and fetches the confirmation receipt; ③ the AV forwards the receipt to the user via communication with the SM and the user renews the anonymous subscriber credential at the PSP using the receipt.

3.3.2 Main Construction

For easy understanding of the construction, we also denote the geo-indistinguishable mechanism on the location-based query data (lat, lon, rng) as the function $\mathcal{DP}(lat, lon, rng, \varepsilon)$,

where lat, lon are coordinates, rng is the query range and ε is the privacy-related parameter, which is similar to [121]. The details will be discussed later.

System Setup

(Offline Setup) the PSP runs the setup algorithm. Bilinear map groups ($\mathbb{G}$, $\mathbb{G}_{\mathbb{T}}$) of a prime order $p > 2^{\lambda}$ are created, where λ is the security parameter and $e(.,.)$ denotes the bilinear map such that $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$. Formally, g is a generator of $\mathbb{G}$ and $e(g, g)$ is defined as g_T. $H : \{0,1\}^* \rightarrow Z_p$, $H' : \{0,1\}^* \rightarrow \mathbb{G}$, and $\hat{H} : Z_p \rightarrow Z_p$ are three cryptographic hash functions, and the PSP's public key is set as $A = g^a$ for a random $a \in Z_p$ and a is the private key. Also, the PSP selects $x, y, z \in Z_p$ and computes $X = g^x$, $Y = g^y$ and $Z = g^z$. $\mu \in Z_p$ is a daily verification key chosen by the PSP. Then, the tuple $\{\mathbb{G}, \mathbb{G}_{\mathbb{T}}, p, g, g_T, e, X, Y, Z, H, H', \hat{H}, \mu, A\}$ is published as the common parameters in the system. Finally, the PSP initializes three empty sets using bloomfilter $\Omega = \{\emptyset\}$, $\Xi = \{\emptyset\}$ and $\Psi = \{\emptyset\}$. Note that, μ, Ω, Ξ and Ψ are reseted per day by the PSP, indicating that the user's anonymous credential is only valid for daily period.

① **PLT Registration**: (1.1) the PLT creates a username and password, and registers itself in the terminal; (1.2) the PLT uploads the identity information, such as the electronic commercial parking lot license, to the PSP, and the PSP verifies the qualification of the parking lot; (1.3) once the verification has been passed successfully, the PLT creates a key pair as $(B = g^b, b)$ where b is chosen randomly over Z_p, calculates the resignature key $R_{ab} = A^{\frac{1}{b}} = g^{\frac{a}{b}}$ and sends the public key B to the PSP; (1.4) the PSP stores B, the parking lot information and completes the registration.

② **User Registration**: (2.1) the user creates a username and password, and registers itself in the user App; (2.2) the user uploads the identity information, such as the electronic driving license, to the PSP, and the PSP verifies the qualification of the user; (2.2) once the verification has been passed successfully, the user finishes the registration.

③ **User Subscription**: (3.1) the user logins into the user App via the valid username and password, and pays the service fee online; (3.2) Once the payment is confirmed by the PSP, the user chooses $(d, r) \in Z_p^2$, constructs $M = Y^d Z^r$, and sends $(M, \hat{H}(d))$ to the PSP; (3.3) the PSP checks whether $\hat{H}(d)$ exists in Ω. If it exists, the PSP guides the user to go back to the step (3.2). Otherwise the PSP adds $\hat{H}(d)$ into Ω; (3.4) the user acts as prover and the PSP as verifier in the non-interactive zero-knowledge proof of knowledge:

$$NIZK\{(d, r)|M = Y^d Z^r\};$$

(3.4) the PSP returns as failure if the proof fails. Otherwise the PSP sends to the user a tuple (W, v), where $v \in Z_p$ and $W = (XM)^{\frac{1}{v+a+\mu}}$; (3.5) the user checks whether $e(W, Ag^{v+\mu}) \stackrel{?}{=} e(XM, g)$. If it fails, the user returns as failure. Otherwise the anonymous credential is stored as $cred = (W, v, d, r)$ locally.

To avert losing the anonymous credential incidentally and support credential recovery, $cred$ is encrypted using a preset secret password $pass$ chosen by the user as $E_{pass}(cred)$, and $E_{pass}(cred)$ can be stored online at the PSP, where $E()$ is a common symmetric encryption algorithm, such as AES.

Service Phase

① **User Authentication**: (1.1) the user acts as prover and the PSP as verifier in the non-interactive zero-knowledge proof of knowledge:

$$NIZK\{(W, v, d, r)|W^{v+a+\mu} = XY^dZ^r\},$$

and logins to the PSP via the App; (1.2) if the proof is successful, the PSP generates a temporary session token $SESS$, and sends it back to the user. Otherwise the PSP returns as failure; (1.3) the user stores the session token $SESS$.

② **Parking Query**: (2.1) the user's current location-based query (lat, lon, rng) is noised by utilizing the geo-indistinguishable mechanism as

$$(lat', lon', rng') = \mathcal{DP}(lat, lon, rng, \varepsilon);$$

(2.2) the user sets the parking requirements and requests the neighbour parking lot information by sending (lat', lon', rng') and $SESS$ to the PSP; (2.3) the PSP filters the parking lots that do not meet the criteria and returns the parking lots list within the query range.

③ **Parking Reservation**: (3.1) the user selects a parking lot from the returned list, sends the reservation request Req to the PSP, where $Req = Info||SESS||Timestamp$ ($Info$ involves the trivial reservation information and $Timestamp$ indicates the current timestamp); (3.2) the user calculates $U = g^{\frac{1}{d+\mu}}$ as the booking token, sends U to the PSP and engages in a non-interactive zero-knowledge proof of knowledge with the PSP, in which the user plays the prover, the PSP plays the verifier:

$$NIZK\{(W, v, d, r)|W^{v+a+\mu} = XY^dZ^r \wedge U = g^{\frac{1}{d+\mu}}\};$$

(3.3) after receiving the request, if the proof succeeds and the token U does not exist in Ξ, the PSP accepts the request and adds U into Ξ. Otherwise the PSP rejects the request;

(3.4) the PSP signs the request as $\sigma = H'(Req)^a$ and relays the request $Req||\sigma$ to the corresponding PLT; (3.5) the PLT verifies the signature of the request by checking

$$e(\sigma, g) \stackrel{?}{=} e(H'(Req), A).$$

If it fails, the reservation request is rejected. Otherwise the PLT generates a unique random string as the temporary parking permit code c, stores it in its local database, and also sends it back to the PSP; (3.6) the PSP signs c as $Sig_c = H'(c||Timestamp||SESS)^a$, stores $SESS$ in its token pool, and gives $c||Sig_c$ back to the user.

Parking Phase

① **Parking Request**: (1.1) the user relays $c||Timestamp||SESS||Sig_c$ and the parking lot information to the AV via the SM; (1.2) the AV switches to the self-driving mode and drives to the selected parking lot according to the received information.

② **Parking Check-In**: (2.1) when connecting to the PLT, the AV sends $c||Timestamp||SESS||Sig_c$ to the PLT; (2.2) the PLT verifies the signature Sig_c by checking

$$e(Sig_c, g) \stackrel{?}{=} e(H'(c||Timestamp||SESS), A).$$

If it is valid, the PLT searches c in its database and assures that whether the AV has already reserved a parking space or not. If c is found in its local database, the PLT deletes c and allows the AV to park inside. Otherwise the PLT returns as failure and refuses to offer the service; (2.3) the PLT re-signs Sig_c by choosing a random $\theta \in Z_p$ as $Sig'_c = (Sig_c^\theta, A^\theta, R_{ab}^\theta)$, and transmits Sig'_c as the confirmation receipt to the AV.

③ **Anonymous Credential Renewal**: (3.1) the AV forwards the receipt Sig'_c to the user's SM and notifies the parking confirmation message on the user's SM; (3.2) After waiting for a random delay, the user applies for a new anonymous credential by sending $c||Timestamp||SESS||Sig'_c||U$ to the PSP; (3.3) after receiving the renewal request, the PSP checks the validity of the credential renewal request by the following three conditions.

- **(Condition.1)** The PSP searches the session token $SESS$ in the session token pool. If $SESS$ exists, the PSP deletes it and this condition is satisfied.
- **(Condition.2)** The PSP verifies the signature Sig'_c by the following equations.

$$e(Sig_c^\theta, g) \stackrel{?}{=} e(A^\theta, H'(c||Timestamp||SESS)),$$

$$e(A^\theta, g) \stackrel{?}{=} e(B, R_{ab}^\theta).$$

  If the equations hold, this condition is satisfied.

- **(Condition.3)** The PSP searches U in Ξ and Ψ. If U exists in Ξ and does not exist in Ψ, the PSP adds U into Ψ and this condition is satisfied.

If any of them are not fulfilled, the PSP rejects the request and returns as failure. Otherwise, the PSP returns with success; (3.4) the user gains a new anonymous credential, following the steps in **User Subscription** except the step (3.1). Moreover, the user can cancel the current parking/reservation session if necessary and perform the above-mentioned steps similarly to gain a new anonymous credential. The difference is that the PSP does not need the parking confirmation message, and must recall the current booking request of the user, according to his/her session token $SESS$ and booking token U.

In addition, to deal with the issue that some important messages, such as the acknowledgment of the parking space, may be lost at the user side accidently, our scheme relies on the PSP as the intermediate servers to store this information. If the users miss the acknowledgment, the PSP can help the user check and download this missing information based on the user's temporary session token. Since the temporary session token is unique and only known by the user and the PSP, only the authorized anonymous user who has already sent this request can check the status of this reservation session. Then, there are two cases: 1) if the request is successful, the user can download the acknowledgment; and 2) if the request is not successful, the user can resend the reservation request.

3.3.3 Protocol Details

Zero-Knowledge Proofs of Knowledge

We present the non-interactive zero knowledge proofs of knowledge ($NIZK$) that are secure in the random oracle model (Fiat-Shamir heuristic).
Proof.I $NIZK\{(d,r)|M = Y^d Z^r\}$:
Prover:
1. Choose $\alpha, \beta \in Z_p$, calculate $\Delta = Y^\alpha Z^\beta$
2. Set $\eta = H(Y, Z, M, \Delta)$
3. Send $(\Delta, M, \hat{\alpha} = d\eta + \alpha, \hat{\beta} = r\eta + \beta)$ to the verifier
Verifier:
1. Calculate $\eta = H(Y, Z, M, \Delta)$
2. Check that $M^\eta \Delta = Y^{\hat{\alpha}} Z^{\hat{\beta}}$
Proof.II $NIZK\{(W, v, d, r)|W^{v+a+\mu} = XY^d Z^r \wedge U = g^{\frac{1}{d+\mu}}\}$:

Note that, the proof can be transformed and rewrited [30] as

$$NIZK\{(v,d,r,\alpha_1,\alpha_2,\beta_1,\beta_2)|W_1 = Y^{\alpha_1}Z^{\alpha_2} \wedge$$
$$1_{\mathbb{G}} = W_1^{-v}Y^{\beta_1}Z^{\beta_2} \wedge U^d = gU^{-\mu} \wedge \frac{e(W_2, Ag^{\mu})}{e(X,g)} =$$
$$e(W_2,g)^{-v}e(Y,g)^d e(Z,A)^{\alpha_1}e(Z,g^u)^{\alpha_1}e(Z,g)^{r+\beta_1}\}$$

where $\alpha_1, \alpha_2 \in Z_p$, $W_2 = WZ^{\alpha_1}$, $\beta_1 = \alpha_1 v$, and $\beta_2 = \alpha_2 v$.
Prover:
1. Choose $\rho_v, \rho_d, \rho_r, \rho_{\alpha_1}, \rho_{\alpha_2}, \rho_{\beta_1}, \rho_{\beta_2} \in Z_p$, calculate $\Delta_1 = Y^{\rho_{\alpha_1}}Z^{\rho_{\alpha_2}}$, $\Delta_2 = W_1^{-\rho_v}Y^{\rho_{\beta_1}}Z^{\rho_{\beta_2}}$, $\Delta_3 = U^{\rho_d}$, $\Delta_4 = e(W_2,g)^{-\rho_v}e(Y,g)^{\rho_d}e(Z,A)^{\rho_{\alpha_1}}e(Z,g^{\mu})^{\rho_{\alpha_1}}e(Z,g)^{\rho_r+\rho_{\beta_1}}$
2. Calculate $\eta = H(X,Y,Z,W_1,W_2,U,\Delta_1,\Delta_2,\Delta_3,\Delta_4)$
3. Send $(\Delta_1,\Delta_2,\Delta_3,\Delta_4,W_1,W_2,U,\hat{\rho_v} = v\eta + \rho_v, \hat{\rho_d} = d\eta + \rho_d, \hat{\rho_r} = r\eta + \rho_r, \hat{\rho_{\alpha_1}} = \alpha_1\eta + \rho_{\alpha_1}, \hat{\rho_{\alpha_2}} = \alpha_2\eta + \rho_{\alpha_2}), \hat{\rho_{\beta_1}} = \beta_1\eta + \rho_{\beta_1}, \hat{\rho_{\beta_2}} = \beta_2\eta + \rho_{\beta_2})$ to the verifier
Verifier:
1. Calculate $\eta = H(X,Y,Z,W_1,W_2,U,\Delta_1,\Delta_2,\Delta_3,\Delta_4)$
2. Check that $W_1^{\eta}\Delta_1 = Y^{\hat{\rho_{\alpha_1}}}Z^{\hat{\rho_{\alpha_2}}}$, $1_{\mathbb{G}}{}^{\eta}\Delta_2 = W_1^{-\hat{\rho_v}}Y^{\hat{\rho_{\beta_1}}}Z^{\hat{\rho_{\beta_2}}}$, $(gU^{-\mu})^{\eta}\Delta_3 = U^{\hat{\rho_d}}$, and $(\frac{e(W_2,Ag^{\mu})}{e(X,g)})^{\eta}\Delta_4 = e(W_2,g)^{-\hat{\rho_v}}e(Y,g)^{\hat{\rho_d}}e(Z,A)^{\hat{\rho_{\alpha_1}}}e(Z,g^{\mu})^{\hat{\rho_{\alpha_1}}}e(Z,g)^{\hat{\rho_r}+\hat{\rho_{\beta_1}}}$

Geo-Indistinguishable Mechanism

Given the parameter $\varepsilon \in \mathbb{R}^+$ (i.e., the default privacy levels can be set as low $\varepsilon = 0.01$, medium $\varepsilon = 0.004$, and high $\varepsilon = 0.001$), and the actual location $pos = (lat, lon) \in \mathbb{R}^2$, the probability density function of noise mechanism (planar Laplacian), on any other point $pos = (lat', lon') \in \mathbb{R}^2$, is $D_{\varepsilon}(pos)(pos') = \frac{\varepsilon^2}{2\pi}e^{-\varepsilon d(pos,pos')}$, where d denotes the Euclidean distance. It can also be represented as polar coordinate model $D_{\varepsilon}(rad,\theta) = \frac{\varepsilon^2}{2\pi}\cdot rad \cdot e^{-\varepsilon\cdot rad}$, where rad and θ are distance and angle with respect to pos. To obfuscate the real location, specifically, θ should be uniformly chosen from $[0, 2\pi)$ and rad should be set as $rad = C_{\varepsilon}^{-1}(p) = -\frac{1}{\varepsilon}(W_{-1}(\frac{p-1}{e}) + 1)$, where W^{-1} is the Lambert W function (the -1 branch) and p should be uniformly chosen from $[0, 1)$. Also, two transformation functions are needed: LatLonToCartesian and CartesianToLatLon, to transform $(lat, lon) \rightarrow (\bar{x}, \bar{y})$ and $(\hat{x}, \hat{y}) \rightarrow (lat', lon')$. Therefore, $\hat{x} = \bar{x} + rad \cdot \cos\theta$ and $\hat{y} = \bar{y} + rad \cdot \sin\theta$. In addition, $rng' = rng - \frac{1}{\varepsilon}(W_{-1}(\frac{\tau-1}{e}) + 1)$, where τ is the accuracy parameter (default $\tau = 0.95$).

Efficient Set Membership Test

The construction requires efficient set membership tests for three sets Ω, Ξ and Ψ, and the standard bloomfilter (BF) data structure is used properly. The characteristics of this data

structure deeply match the requirements of our construction, which include the compressed storage for large dataset, the zero false negative rate, and the fast search algorithm: since the number of reservation/parking requests is large, the BF helps diminish the storage overheads; since each booking token U can only be used for one time, it could not be missed by BF if it had been used due to the zero false negative rate; the fast search algorithm can accelerate the testing speed and reduce the computational costs. Generally, a BF consists of an array of m cells, each of which is a bit with an initial value 0, and k independent random hash functions, where m and k are determined by the maximum number of data items supported by BF and the false positive ratio of BF.

3.4 Privacy and Security Analysis

3.4.1 Privacy Analysis

Following the privacy requirements discussed earlier, our analysis will focus on how the proposed scheme can ensure the user's pseudonymity, unlinkability and geo-indistinguishability.

Pseudonymity: each user has totally different anonymous credentials (W, v, d, r) for different reservation/parking sessions in our proposed scheme. The anonymous credential, as a unique pseudonym defined by the user and confirmed by the PSP (Proof.I), can be verified by the PSP as the valid anonymous credential (part of Proof.II) during the anonymous authentication process. Hence, the user's pseudonymity relies on the security of two zero-knowledge proof protocols. Specifically, Proof.I is an adapted version of the CL signature scheme [122] and Proof.II is an adapted version of the BBS/BBS+ signature schemes [23, 30]. Their security proofs are thus relatively straightforward.

Unlinkability: the PSP can perform the pseudonym linking attack, and our scheme guarantees that the possibility that the PSP succeeds in linking one user's two reservation/parking sessions cannot be better than guessing. In other words, the PSP cannot link the user's real identity and the user's first anonymous credential during user subscription, and the PSP cannot link the user's previous anonymous credential and renewed anonymous credential during anonymous credential renewal. This property of unlinkability is dependent on two zero-knowledge proof protocols Proof.I and Proof.II. When the user applies for the anonymous credential using his/her real identity, the PSP only knows that the registered user acquires a valid anonymous credential, it does not know the values of (d, r) but can still acknowledge the anonymous credential (W, d, v, r) (Proof.I) as a valid BBS+ signature. During parking reservation, the user's reservation token U cannot be

linked to a specific anonymous credential by the PSP since the PSP does not have d. Similarly, the PSP only knows that a new anonymous credential is generated and assigned to the anonymous user during the renewal period, but it does not know the content of this new credential. During anonymous authentication, the PSP and the user run a non-interactive zero knowledge proof to verify the BBS/BBS+ signature, i.e., the PSP can verify $W^{v+a+\mu} = XY^dZ^r$ without knowing the values of (W, d, v, r), which guarantees the unlinkability.

Geo-indistinguishability: ε-geo-indistinguishability is defined as $\frac{P(Z|x)}{P(Z|x')} \leq e^{\varepsilon d(pos,pos')}$, where P is the conditional probability. Each observation is $Z \subseteq \mathcal{Z}$, where $\mathcal{Z}$ is a set of possible reported locations, and $d(pos, pos')$ is the Euclidean distance between pos and pos'. By adding a planar laplacian noise $\mathcal{N} = (rad, \theta)$ to the original location (lat, lon) in the proposed scheme, the reported location can be viewed as an obfuscated location $pos' = (lat', lon')$, and the ε-geo-indistinguishabilitys is satisfied. The detailed proof can be found in [117].

3.4.2 Security Analysis

We focus on how the proposed scheme can be resilient to the *"Double-Reservation Attack"* in the security analysis. The proposed scheme is designed based on the idea of generating one-time booking token for each registered user and his/her every booking/parking session. To prevent the attack, the fundamental intuition is to make sure that each anonymous user should and must have only one valid token at one time. In specific, each user can obtain the token in two stages: user subscription and anonymous credential renewal. The PSP can easily assure that each registered user only applies for one anonymous credential during user subscription. If the user has been allocated the anonymous credential, other similar requests will be dropped since the account information will be recorded. For the renewal process, the situation becomes complex but can still be addressed based on three decision conditions:

- **(Condition.1)** The renewal request comes from a current reservation/parking session by checking the session token $SESS$.
- **(Condition.2)** The verification of a PLT's confirmation receipt guarantees that the anonymous user's parking session is accomplished by checking the signature Sig'_c.
- **(Condition.3)** The booking token U has already been used for booking and has not been used for renewing by performing set membership tests in Ξ and Ψ. Proof.II

indicates that the token U is authenticated by the PSP, i.e., the token cannot be forged.

With the aforementioned three conditions, the PSP can update the user's exclusive anonymous credential. Namely, the attack has been prevented. The abnormal timestamp information for each reservation/parking session (i.e., time duration between reservation and parking is too short) may help detect the suspicious PLT who may collude with the malicious user even though this collusion attack gains no benefit for the attackers.

In addition, since the parking reservation is a paid service, the proposed scheme also guarantees that only the premium users who paid the fees can use this service. The daily verification key μ, included in each user's anonymous credential and reservation token, makes sure that each user needs to refresh his/her subscription information everyday. If the subscription is expired, he/she will not be allowed to apply for a valid anonymous credential.

3.5 Performance Evaluation and Implementation

In this section, we evaluate the performance of the proposed scheme in terms of communication overheads, computational and storage costs. Also, a WiFi-based testbed has been built to further demonstrate the scheme's practicality.

3.5.1 Simulation Settings

Table 3.2: Testbed setting

Role	Machine	Hardware and Software
PSP	Workstation	Intel i7-6700K @ 4.00 GHZ; 32 GB memory; Windows 10
PLT	Notebook	Intel Core i5-7200U @ 2.60 GHz; 16 GB memory; Windows 10
AV	Galaxy S4	1.9 GHz Krait 300; 2 GB memory; Android 5.0
Smartphone	Galaxy S4	1.9 GHz Krait 300; 2 GB memory; Android 5.0

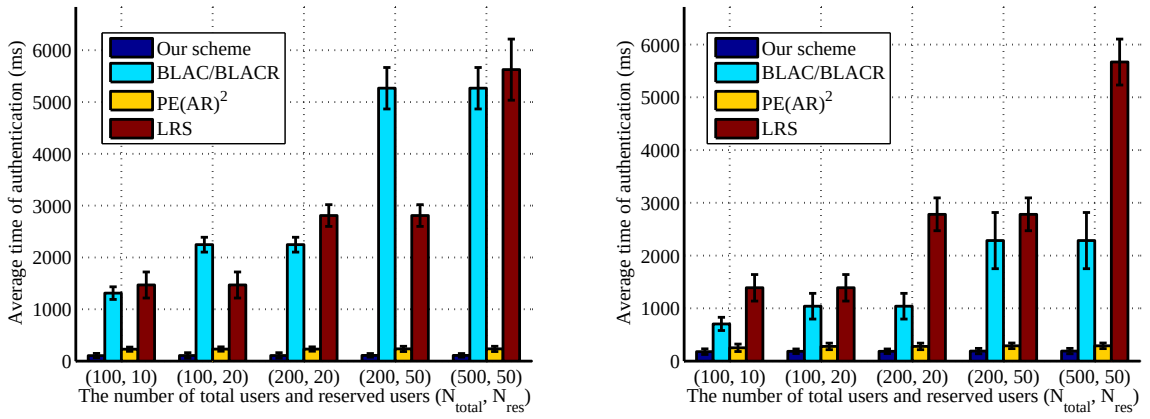
In our simulation, we compare our solution with three traditional solutions based on the blacklistable anonymous credential (BLAC/BLACR) [114], the blacklistable anonymous credential with universal accumulator (PE(AR)^2) [113] and linkable ring signature (LRS) [115], which could also deal with the " Double-Reservation Attack" anonymously under some conditions. This simulation is built on a JAVA-based simulator and conducted on a notebook with Intel Core i5-7200U CPU @ 2.60GHz and 16.00 GB memory. Then, we test the scheme's performance in a testbed of one workstation, one notebook and one Android phone. These machines play the roles of the PSP, the PLT, the AV and smartphone, respectively. The hardware and software of these machines are shown in Table 3.2.

3.5.2 Performance Comparisons

Since our solution is particularly proposed for the AVP system, it has many characteristics which the previous protocols do not have (e.g., location obfuscation at user side and the participation of PLT). Hence, we mainly investigate performance comparisons of the **anonymous authentication costs** (i.e., **the costs of parking reservation**), which involves the communication overheads, computational and storage costs. For the BLAC/BLACR-based solution, each user owns a anonymous credential after finishing payment, and the PSP maintains an anonymous blacklist. When a user reserves a vacant parking space via the PSP, he/she has to prove to the PSP (one by one) that he/she is not shown on that anonymous blacklist. When the reservation is finished, his/her anonymous credential is added to the blacklist to prevent the "Double-Reservation Attack". For the PE(AR)^2-based solution, the procedure is similar to that of the BLAC/BLACR-based solution, while the difference is that the proof between the user and the PSP is designed based on a universal accumulator to improve the computational efficiency for both sides. For the LRS-based solution, each user owns a unique ring signature to represent his/her identity in a pre-defined group. When a user books a vacant parking space via the PSP, he/she has to generate a ring signature, which indicates that he/she is from this group but conceals the specific identity, and submits this signature to the PSP. When the reservation is finished, his/her current reservation request can be linked by the PSP to the future requests to identify whether these two requests come from the same user in the group anonymously.

We use the BouncyCastle library and JAVA Pairing-Based Cryptography (JPBC) library to implement the cryptographic building blocks in our simulator. The elliptic curve of the bilinear pairing is chosen with a base field size of 512 bits and the order p is 160 bits. To keep the consistency, the simulation is conducted under the same setting. The number of total users N_{total} is set as $\{100, 200, 500, 10000\}$, and the number of reserved users (the user has finished the reservation but not achieved parking yet) N_{res} is set as $\{10, 20, 50, 1000\}$

in our simulation. The numerical results of computational costs are shown in Fig 3.5, and the results are averaged by 100-times simulations.



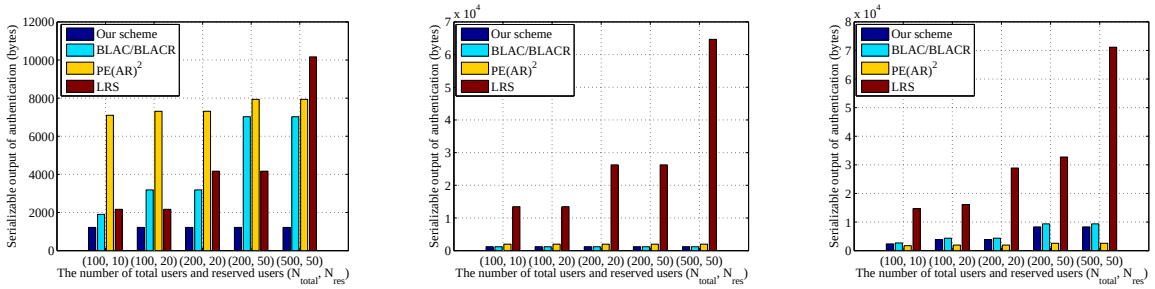
(a) Computational costs at user side with different number of users

(b) Computational costs at PSP side with different number of users

Figure 3.5: Computational costs compared with the existing schemes

Apparently, the execution time of BLAC/BLACR-based and LRS-based solutions are linearly increased with the growth of N_{res} and N_{total} respectively, but the running time of our scheme and PE(AR)^2-based solution is not impacted by either of them (i.e., our scheme's execution time is almost fixed 110 ms and 180 ms at user side and PSP side, and the PE(AR)^2-based solution's execution time is almost fixed 240 ms and 260 ms at user side and PSP side). The reason is that, our scheme just requires the user to provide a one-time reservation token during each anonymous authentication process which is very efficient. However, since BLAC/BLACR requires each user to retrieve the whole blacklist and to prove to the PSP separately that he/she does not exist in that list, the proof should be executed N_{res} times between the user and the PSP (i.e., the running time of BLAC/BLACR is almost 101.211 s and 41.332 s at user side and PSP side when $N_{res} = 1000$). In another way, LRS requires each user to sign the signature on behalf of the whole group to preserve the anonymity, which indicates that the signature should involve N_{total} group member information (i.e., the execution time of LRS is almost 147.429 s and 146.873 s at user side and PSP side when $N_{total} = 10000$) and cannot be distinguished by the PSP. Although the PE(AR)^2-based solution is almost equally efficient as our scheme, it still costs more time because the user has to re-generate the accumulated witness and perform a more complex proof on during each anonymous authentication, which are not necessary in our

scheme. We also give the error bars which indicate the time out and help the users to determine whether they have lost the messages to some extent. In addition, we compare the communication overheads and storage costs among our scheme, the BLAC/BLACR, the PE(AR)^2 and the LRS, and the numerical results are shown in Figure 3.6. Note that, the communication overheads (uplink and downlink) and storage costs are the serializable output as the byte array type in our JAVA-based simulator, and may be different from other programming languages due to diverse data types.



(a) Communication overheads between user and PSP (uplink and downlink) with different number of users

(b) Storage costs at user side with different number of users

(c) Storage costs at PSP side with different number of users

Figure 3.6: Communication overheads and storage costs compared with the existing schemes

The Figure 3.6 (a) shows that our scheme only requires around 1200-byte bandwidth per request but both the BLAC/BLACR and LRS needs more than 2000-byte bandwidth even if there are only 100 users and 10 reserved users in the system. Especially, the PE(AR)^2-based solution requires more than 7000-byte bandwidth during anonymous authentication since it has five zero knowledge proofs for every request. When $N_{total} = 10000$ and $N_{res} = 1000$, the bandwidth requirements are significantly large (i.e., each user uploads 295275 bytes for the BLAC/BLACR, 27884 bytes for the PE(AR)^2, and 200129 bytes for the LRS). Here, the BLAC/BLACR-based solution needs more bandwidth than the LRS-based solution because the user has to download the newest blacklist before any authentication takes place, and the blacklist changes as long as the parking reservation happens. Hence, the blacklist downloading overheads cannot be avoided. However, the PE(AR)^2 has a better performance than the BLAC/BLACR and the LRS when N_{total} and N_{res} is large. The reason is that the users can download the whole newest blacklist in an accumulator for the PE(AR)^2-based solution, which fills the gap of BLAC/BLACR.

For the BLAC/BLACR and PE(AR)^2, the PSP stores the blacklist and its private key,

and the user stores the anonymous credential. The storage costs of LRS are decided by the number of group members (users). If there are more group members, the user has to store not only his/her key pairs but also other member's public keys, and the PSP needs to store all group members' public keys and reserved users' signatures. In our scheme, the user stores the anonymous credential, and the PSP stores its private key and three sets (i.e., the efficient bloomfilter data structure is not considered in the comparison for the sake of fairness). Although the storage costs of our scheme are not the best one compared to that of the previous solutions, the Figure 3.6 (b) and (c) show that the costs are still small enough to support scalability. In the real world, there may exist more than 10000 users and 1000 reserved users in the system, the storage costs of our scheme are also acceptable (1205 bytes and 148889 bytes at user side and PSP side). The BLAC/BLACR's costs are 1185 bytes and 167724 bytes, the PE(AR)^2 costs are 1973 bytes and 22529 bytes, and the LRS's costs are 1280653 bytes and 1408741 bytes at user side and PSP side.

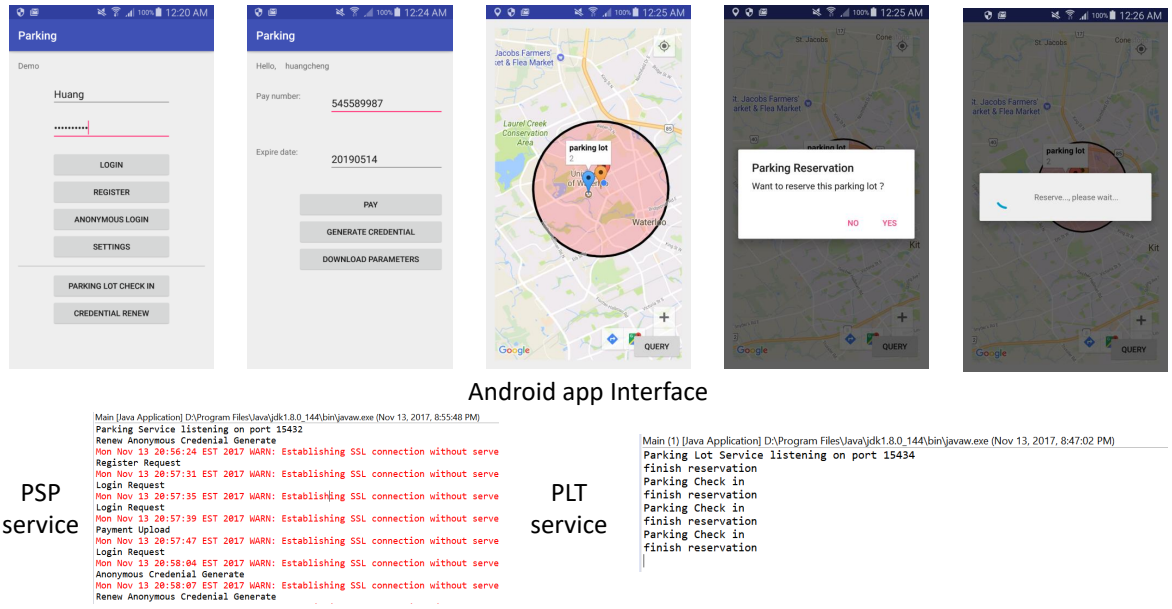


Figure 3.7: Selected interfaces of user, server, and terminal

3.5.3 Implementation on Testbed

The PSP, PLT and the smartphones are connected via WiFi, and the communication among them is designed based on the JAVA socket programming. For simplicity, the automated vehicle and smartphone at user side are programmed into one android application, while the PSP and PLT own separated JAVA server applications, which support multiple threads. The information of registered users and PLTs are store in the MySQL database which is deployed at PSP side. As shown in Figure 3.7, the android application supports basic functions, such as user registration, user login, user subscription (after user login) anonymous login (i.e., user authentication), parking query (after anonymous login), parking reservation (after parking query), parking check-in (including parking request) and anonymous credential renewal. As a research demo, just one PLT application is deployed with the fixed information near the University of Waterloo, and a single PLT registration application is developed, but it is still enough to test the performance of our scheme since multiple PLTs will not impact the performance from a design standpoint. The test results are shown in Table 3.3. Most of the delays are measured from the android client side, starting from the request generation to the operation completion. The most time-consuming operation of our scheme is the parking reservation which costs almost 3 seconds. User subscription, user authentication and anonymous credential renewal cost around 2 seconds, while other operations cost less than 300 ms. Therefore, our scheme is very efficient in the WiFi-based testbed.

Table 3.3: The performance (delay) of our testbed

Setup Phase		Service Phase		Parking Phase	
Subphase	Time	Subphase	Time	Subphase	Time
PLT Registration	$\approx$ 300 ms	User Authentication	$\approx$ 2 s	Parking Request	$\approx$ 100 ms
User Registration	$\approx$ 100 ms	Parking Query	$\approx$ 100 ms	Parking Check-In	$\approx$ 150 ms
User Subscription	$\approx$ 2 s	Parking Reservation	$\approx$ 3 s	Anonymous Credential Renewal	$\approx$ 2 s

3.6 Summary

In this chapter, we have proposed a privacy-preserving reservation scheme for securing AVP system. The security model has been first presented to define the privacy requirements and the potential attacks in this system. Then, the proposed scheme has been designed particularly based on the features of AVP system, to guarantee both the user's identity privacy and location privacy, and prevent the "Double-Reservation Attack" performed by the malicious users. Note that, the vacant parking spaces are chosen by the drivers themselves, which makes the location privacy of any driver can be easily protected by location obfuscation mechanisms.

Chapter 4

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

4.1 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 [123, 124]. 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 [125]. 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 [126]. 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 [127, 128]. 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 standpoint [129, 130, 131], 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 [132, 133, 134, 135]. 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 [136]. 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 [86, 88, 137, 138]. The most related work is SePCAR [86], 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) [139]. 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

[1] `https://www.enterprisecarshare.ca/`
[2] `https://www.car2go.com/`
[3] `https://www.zipcar.ca/`

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

4.2 Models and Design Goals

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

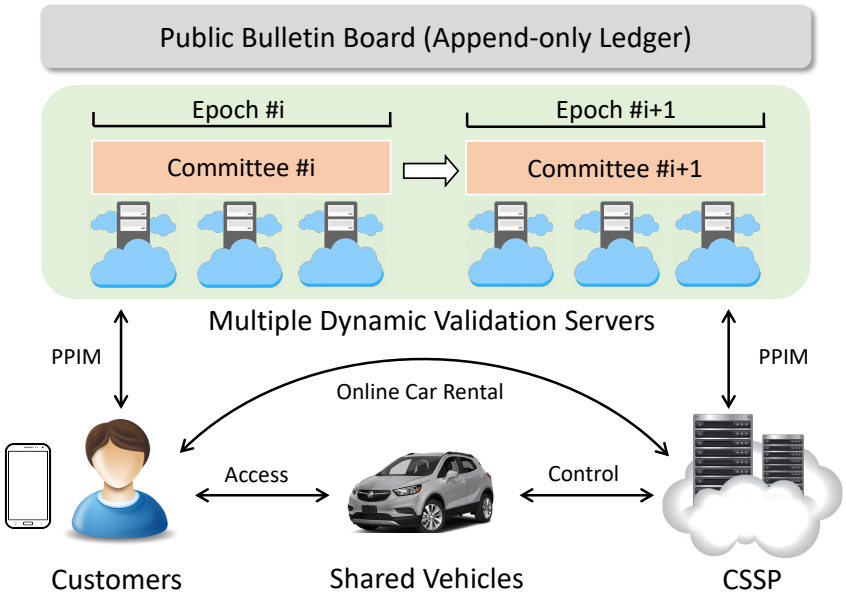


Figure 4.1: System model

4.2.1 System Model

In the system model, there exist five entities in a car sharing service, as shown in Figure 4.1: 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 dynamically change 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 [130], where other entities can read/write the data. It can be regarded as a public bulletin board [140].

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.

4.2.2 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.

4.2.3 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.

4.3 Proposed DAPA

In this section, we propose a decentralized, accountable, and privacy-preserving car sharing architecture, named DAPA. We begin with present an overview of DAPA. Then, we propose a privacy-preserving identity management (PPIM) scheme, serving as a basic module for DAPA. Finally, we show the detailed construction of DAPA.

4.3.1 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 [141] 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.

4.3.2 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 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 4.1 shows the notations frequently used in PPIM.

Concretely, PPIM involves six parts: parameters generation, identity registration, identity hiding, identity transferring, identity recovery, and identity revocation.

• *Parameter Generation (PGen).* This part is run by CSSP during system setup. CSSP can generate parameters as follows: i) $\tilde{p}$ is a $\tilde{l}$-bit prime number that satisfies $\tilde{p} = 2\tilde{q} + 1$ and $\tilde{q}$ is also a prime number; ii) $\tilde{G}$, $\mathfrak{G}$ 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; iii) $(\tilde{g}, \tilde{h})$ are two generators of $\tilde{G}$ and $\mathfrak{g}$ is a generator of $\mathfrak{G}$; iv) τ is a random number picked from $Z^*_{\tilde{p}}$ and $\bar{G} \subset Z^*_{\tilde{p}}$ is a cyclic group of prime order $\tilde{q}$; v) H, H' are two cryptographic hash functions: $H : \{0,1\}^* \times \{0,1\}^* \rightarrow \{0,1\}^l$ and $H' : \{0,1\}^* \rightarrow \{0,1\}^{\tilde{l}}$; vi) its private/public key pair is $(y, Y = \mathfrak{g}^y)$, where $y \in_R Z^*_{\tilde{p}}$; vii) a upper bound $\tilde{N}$ for revoked customers and public information $\vec{\Phi} = \{\Phi_1 = \mathfrak{g}^{\tau}, \Phi_2 = \mathfrak{g}^{\tau^2}, ..., \Phi_{\tilde{N}} = \mathfrak{g}^{\tau^{\tilde{N}}}\}$.

• *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

Table 4.1: Notations frequently used in PPIM

Notation	Definition
$\tilde{G}$, $\mathfrak{G}$, G_T	three groups that support bilinear maps
$\tilde{l}$	security parameter
$\tilde{p}, \tilde{q}$	two primes satisfy $\tilde{p} = 2\tilde{q} + 1$
$H'()$	a cryptographic hash function
$\tilde{g}, \tilde{h}$	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
$(\tilde{y}, \tilde{Y})$	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)	ciphertext of σ

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 key as $\tilde{y} \in_R Z_{\tilde{p}}^*$ and generates the public keys as $\tilde{Y} = \tilde{g}^{\tilde{y}}$, and writes the public keys $\tilde{Y}$ into the bulletin board.

- The customer distributes the private key $\tilde{y}$ to N VSs (current committee members) $\mathfrak{P} = (P_1, P_2, ..., P_N)$ 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 $\tilde{y}$, 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 $f(x) = a_0 + a_1 x + ... + a_{t-1} x^{t-1}$ over $Z_{\tilde{p}}^*$ of degree $t-1$. The customer sets $f(0) = a_0 = \tilde{y}$. 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 $\{\tilde{g}^{s_k}\}_{k=1}^{N}$ and $\{D_k = \tilde{g}^{a_k}\}_{k=1}^{t-1}$ into the bulletin board.
  * (Complaining Stage) P_i verifies the shares it received from the customer. P_i checks if

$$\tilde{g}^{s_i} = \tilde{Y} \cdot \prod_{k=1}^{t-1} (D_k)^{i^k}. \tag{4.1}$$

    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. (4.1) 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. (4.1). 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 $S_{inv} = \{\sigma_1, \sigma_2, ..., \sigma_{n'}\}$. The customer generates a polynomial $\bar{f}(x) = \prod_{i=1}^{n'} \eta_i x^i$ satisfies $\prod_{\sigma' \in S_{inv}} (\sigma' + x) = \bar{f}(x) \cdot (\sigma + x) + d$, where η_i is the coefficient of the polynomial $f(\rho)$, n' is the size of S_{inv}, and d is a constant. The customer sets the witness (a, d), where $a = \tilde{g}^{f(\eta_i)} = \prod_{i=1}^{n'} \Phi_i^{\eta_i}$.

- The customer chooses a random number $r \in_R Z_{\tilde{p}}^*$ and encrypts her identity credential σ as (u, w):

$$u = \tilde{g}^r, w = \tilde{g}^{\sigma} \tilde{Y}^r.$$

Note that, this is a ciphertext of an Elgamal encryption that provides chosen-plaintext security.

- The customer generates a non-interactive zero-knowledge proof π, which proves three properties: 1) (u, w) is a valid ciphertext that is encrypted under the public key $\tilde{Y}$; 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 is a random number picked from $Z_{\tilde{p}}^*$.

$$\begin{aligned} & NIZK\{(r, \sigma, \gamma, a, d) : \\ & u = \tilde{g}^r \wedge w = \tilde{g}^{\sigma}\tilde{Y}^r \wedge C = \tilde{g}^{\frac{\gamma}{y+\sigma}} \\ & \wedge e(a, \mathfrak{g}^{\sigma}\Phi_1)e(\tilde{g}, \mathfrak{g})^d = e(c, \mathfrak{g}) \wedge d \neq 0\}. \end{aligned} \tag{4.2}$$

- 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) 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 key $\tilde{y}$) 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 $\hat{f}_i(x) = \hat{a}_{i,0} + \hat{a}_{i,1}x + ... + \hat{a}_{i,\hat{t}-1}x^{\hat{t}-1}$ over $Z_{\tilde{p}}^*$ of degree $\hat{t} - 1$. P_i sets $\hat{f}_i(0) = \hat{a}_{i,0} = s_i$ and writes $\hat{C}_{i,k} = \tilde{g}^{\hat{a}_{i,k}}$ 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

$$\tilde{g}^{\hat{s}_{i,j}} = \prod_{k=0}^{\hat{t}-1} (\hat{C}_{i,k})^{j^k}. \tag{4.3}$$

  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. (4.3) 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. (4.3); ② P_j tests whether $\hat{C}_{i,0}$ is equal to $\tilde{g}^{s_i}$. 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 $\tilde{g}^{s_j}$ 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), 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 $\tilde{g}^{s_i}$ (or $\tilde{g}^{s_j}$). After receiving t (or $\hat{t}$) secret shadows, the secret $\tilde{y}$ can be recovered by CSSP as follows.

$$\tilde{y} = \sum_{i=1}^{t} (s_i \cdot \prod_{k=1, k \neq i}^{t} \frac{k}{k-i}). \tag{4.4}$$

To decrypt the identity of the customer, CSSP checks $\tilde{Y} = \tilde{g}^{\tilde{y}}$ and computes $\tilde{g}^{\sigma} = w \cdot u^{-\tilde{y}}$. Finally, by searching all registered users, the CSSP can easily derive σ, 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.

4.3.3 Detailed Construction of DAPA

• **System Setup**: During this phase, CSSP runs *PPIM.PGen* to generate the public cryptographic parameters $Params$ and the private key $y \in Z_{\tilde{p}}^*$.

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

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

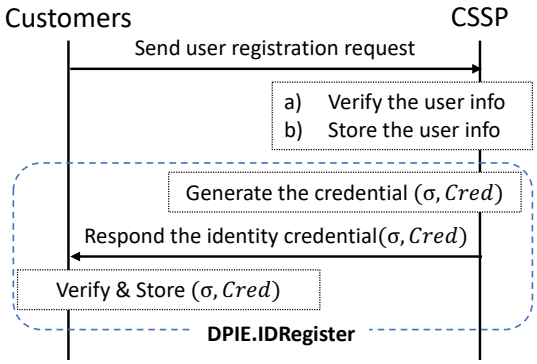


Figure 4.2: Customer registration procedure

• **Customer Registration**: As shown in Figure 4.2, a customer registers herself at CSSP. Following the protocol *PPIM.IDRegister*, the customer sends a unique username/password (Note that, they are used for re-generating or 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**: As shown in Figure 4.3, a customer achieves online car rental and obtains a code to access the shared car through CSSP and VSs. Following the protocol *PPIM.IDHide*, the customer generates the key pairs $\{(\tilde{y}, \tilde{Y}\}$ and publishes the public keys $\tilde{Y}$ to VSs. The customer then shares the private key $\tilde{y}$ 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), 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 $\{ID_{user}, Key_{user}, (u, w), \pi\}$ 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., Boneh–Lynn–Shacham signature [22]), 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

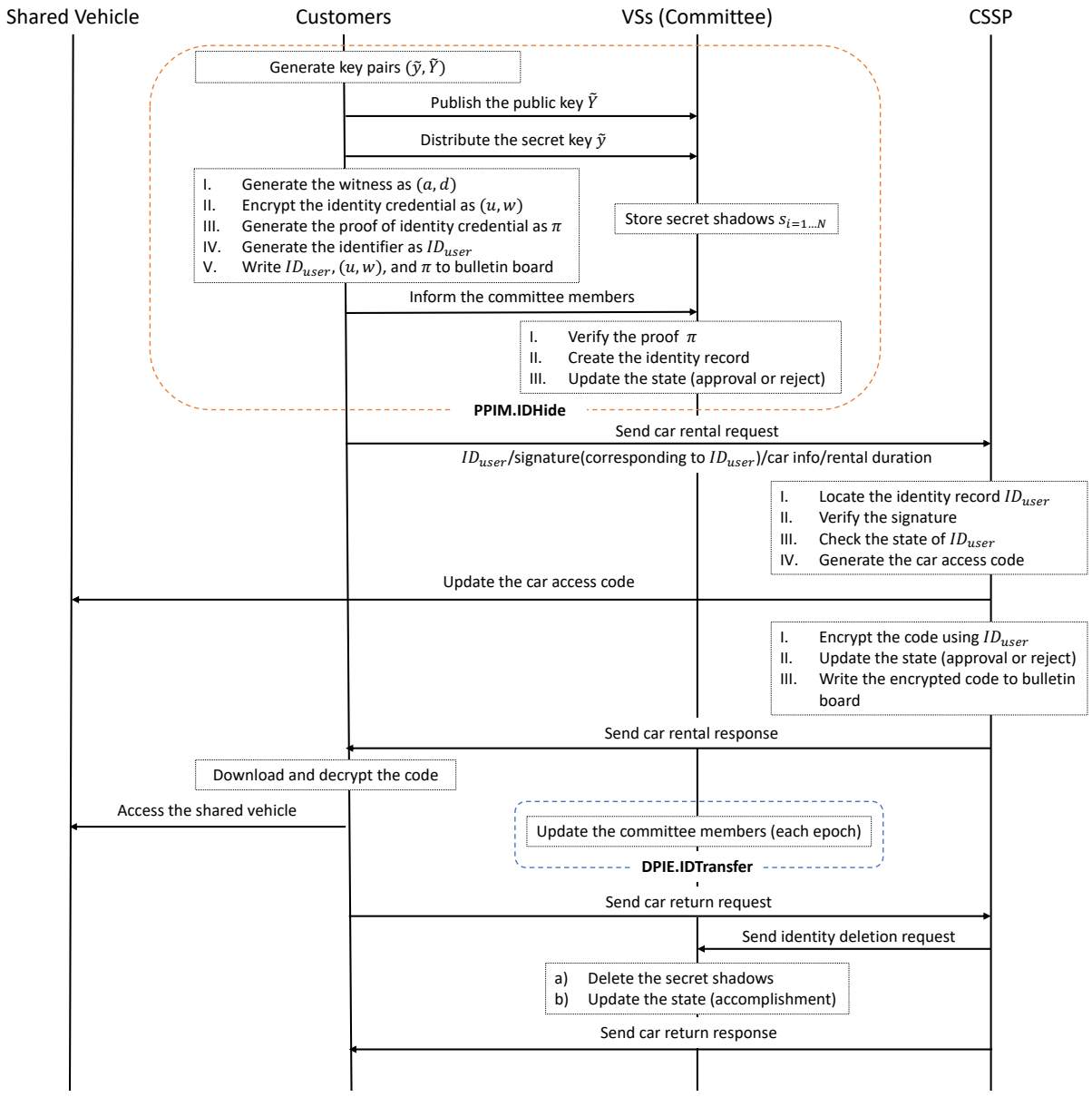


Figure 4.3: Car rental procedure

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.

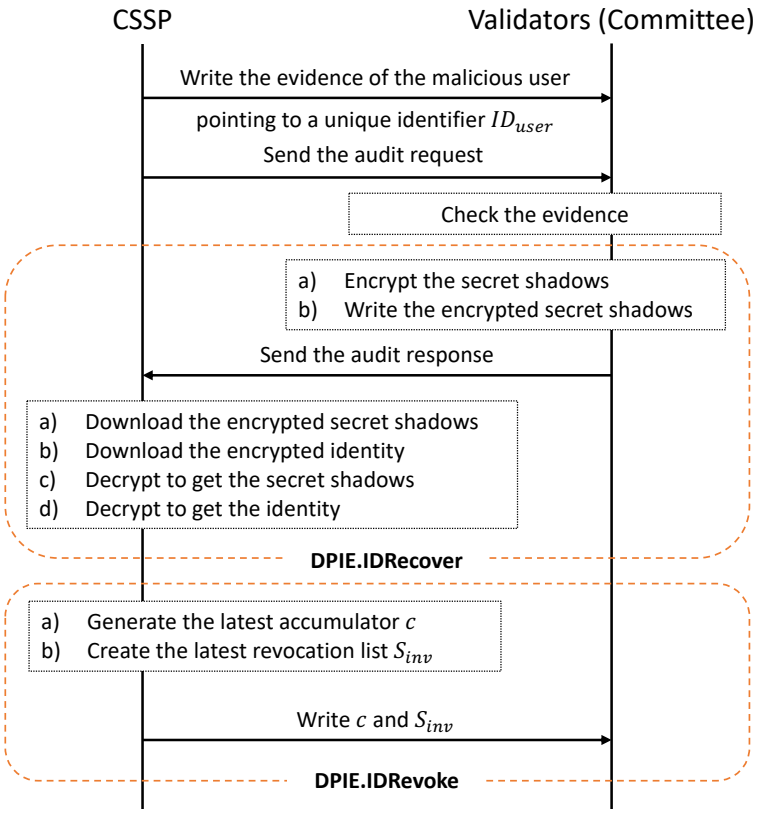


Figure 4.4: Car audit and customer revocation procedure

• **Car Audit**: As shown in Figure 4.4, 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**: As shown in Figure 4.4, 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.

4.4 Security Analysis

In this section, we first analyze the proposed NIZK, i.e., Eq. (4.2), 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.

4.4.1 Security Analysis of NIZK

Lemma 1. Eq. (4.2) 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 some auxiliaries as $C = Cred^{\gamma}$, $\theta_1 = \sigma\beta_1$, $\theta_2 = \sigma\beta_2$, $\theta_3 = d\beta_3$, $\theta_4 = d\beta_4$, $B_1 = \tilde{g}^{\beta_1}\tilde{h}^{\beta_2}$, $B_2 = a\tilde{h}^{\beta_1}$, $B_3 = \tilde{g_1}^{\beta_3}\tilde{g}_2^{\beta_4}$, $B_4 = \tilde{g}_3^{\theta_3}$, where $\beta_1, \beta_2, \beta_3, \beta_4$ are chosen from $Z_{\tilde{p}}^*$. Then, the proof can be transformed into another (equivalent) proof as follows.

$$
\begin{aligned}
&NIZK\{(r, \sigma, \gamma, d, \beta_1, \beta_2, \beta_3, \beta_4, \theta_1, \theta_2, \theta_3, \theta_4) : \\
&u = \tilde{g}^r \wedge w = \tilde{g}^{\tilde{\sigma}}\tilde{Y}^r \wedge \frac{e(B_2, \Phi_1)}{e(c, \mathfrak{g})} = e(\tilde{g}, \mathfrak{g})^{-d} e(\tilde{h}, \mathfrak{g})^{\theta_1} e(\tilde{h}, \Phi_1)^{\beta_1} e(B_2, \mathfrak{g})^{-\sigma} \\
&\wedge C = g^{\frac{\gamma}{y+\sigma}} \wedge B_1 = \tilde{g}^{\beta_1}\tilde{h}^{\beta_2} \wedge 1 = B_1^{-\sigma}\tilde{g}^{\theta_1}\tilde{h}^{\theta_2} \\
&\wedge B_3 = \tilde{g_1}^{\beta_3}\tilde{g}_2^{\beta_4} \wedge 1 = B_3^{-d}\tilde{g_1}^{\theta_3}\tilde{g}_2^{\theta_4} \wedge B_4 = \tilde{g}_3^{\theta_3} \wedge d \neq 0\}.
\end{aligned}
$$

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

$$\begin{aligned}
&\dot{u} = g^{\dot{r}}, \dot{w} = \tilde{g}^{\dot{\sigma}} \tilde{Y}^{\dot{r}}, \dot{C} = e(C, \mathfrak{g})^{-\dot{\sigma}} e(\tilde{g}, \mathfrak{g})^{\dot{\gamma}}, \\
&\dot{B_{1,1}} = \tilde{g}^{\dot{\beta_1}} \tilde{h}^{\dot{\beta_2}}, \dot{B_{1,2}} = B_1^{-\dot{\sigma}} \tilde{g}^{\dot{\theta_1}} \tilde{h}^{\dot{\theta_2}}, \\
&\dot{B_{3,1}} = \tilde{g_1}^{\dot{\beta_3}} \tilde{g}_2^{\dot{\beta_4}}, \dot{B_{3,2}} = B_3^{-\dot{d}} \tilde{g_1}^{\dot{\theta_3}} \tilde{g}_2^{\dot{\theta_4}}, \dot{B}_4 = \tilde{g}_3^{\dot{\theta_3}}, \\
&\dot{D} = e(\tilde{g}, \mathfrak{g})^{-\dot{d}} e(\tilde{h}, \mathfrak{g})^{\dot{\theta_1}} e(\tilde{h}, \mathfrak{g})^{\tau \dot{\beta_1}} e(B_2, \mathfrak{g})^{-\dot{\sigma}}.
\end{aligned}$$

Afterwards, the customer calculates the challenge $ch = H'(\tilde{g}, u, w, B_1, B_2, B_3, B_4, C, \dot{u}, \dot{w}, \dot{B}_1, \dot{B}_2, \dot{B}_3, \dot{B}_4, \dot{C}, \dot{D})$, $\ddot{r} = \dot{r} - ch \cdot r$, $\ddot{\sigma} = \dot{\sigma} - ch \cdot \sigma$, $\ddot{\gamma} = \dot{\gamma} - ch \cdot \gamma$, $\ddot{d} = \dot{d} - ch \cdot d$, $\ddot{\beta_1} = \dot{\beta_1} - ch \cdot \beta_1$, $\ddot{\beta_2} = \dot{\beta_2} - ch \cdot \beta_2$, $\ddot{\beta_3} = \dot{\beta_3} - ch \cdot \beta_3$, $\ddot{\beta_4} = \dot{\beta_4} - ch \cdot \beta_4$, $\ddot{\theta_1} = \dot{\theta_1} - ch \cdot \theta_1$, $\ddot{\theta_2} = \dot{\theta_2} - ch \cdot \theta_2$, $\ddot{\theta_3} = \dot{\theta_3} - ch \cdot \theta_3$, $\ddot{\theta_4} = \dot{\theta_4} - ch \cdot \theta_4$. The customer finally sends the proof as follows.

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

After receiving the proof, VS as the verifier computes ch and verifies the proof by checking whether the following relations hold.

$$\begin{aligned}
&\dot{u} = u^{ch} g^{\ddot{r}}, \dot{w} = w^{ch} \tilde{Y}^{\ddot{r}} \tilde{g}^{\ddot{\sigma}}, \\
&\dot{C} = e(C, Y)^{ch} e(C, \mathfrak{g})^{-\dot{\sigma}} e(\tilde{g}, \mathfrak{g})^{\ddot{\gamma}}, B_4 \neq 1, \\
&\dot{B_{1,1}} = B_1^{ch} \tilde{g}^{\ddot{\beta_1}} \tilde{h}^{\ddot{\beta_2}}, \dot{B_{1,2}} = 1^{ch} B_1^{-\ddot{\sigma}} \tilde{g}^{\ddot{\theta_1}} \tilde{h}^{\ddot{\theta_2}}, v = abs(v), \\
&\dot{B_{3,1}} = B_3^{ch} \tilde{g_1}^{\ddot{\beta_3}} \tilde{g}_2^{\ddot{\beta_4}}, \dot{B_{3,2}} = 1^{ch} B_3^{-\ddot{d}} \tilde{g_1}^{\ddot{\theta_3}} \tilde{g}_2^{\ddot{\theta_4}}, \dot{B}_4 = B_4^{ch} \tilde{g}_3^{\ddot{\theta_3}}, \\
&\dot{D} = (\frac{e(B_2, \Phi_1)}{e(c, \mathfrak{g})})^{ch} e(\tilde{g}, \mathfrak{g})^{-\ddot{d}} e(\tilde{h}, \mathfrak{g})^{\ddot{\theta_1}} e(\tilde{h}, \Phi_1)^{\ddot{\beta_1}} e(B_2, \mathfrak{g})^{-\ddot{\sigma}}.
\end{aligned}$$

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.,

$$\begin{aligned}
\{&C, B_1, B_2, B_3, B_4, u, w, ch, ch', \ddot{r}, \ddot{r}', \ddot{\sigma}, \ddot{\sigma}', \\
&\ddot{\gamma}, \ddot{\gamma}', \ddot{d}, \ddot{d}', \ddot{\beta}_1, \ddot{\beta_1}', \ddot{\beta}_2, \ddot{\beta_2}', \ddot{\beta}_3, \ddot{\beta_3}', \ddot{\beta}_4, \ddot{\beta_4}', \ddot{\theta}_1, \ddot{\theta_1}', \\
&\ddot{\theta}_2, \ddot{\theta_2}', \ddot{\theta}_3, \ddot{\theta_3}', \ddot{\theta}_4, \ddot{\theta_4}'\}.
\end{aligned}$$

The first transcript is the original transcript in the proof while the second transcript is different from the first one, e.g. $\ddot{r}' = \dot{r} - ch' \cdot r$. The only difference is that the challenges ch and ch' is different. If VS accepts both transcripts, the witnesses for the statement in Eq. (4.2) can be computed and extracted by the following equations.

$$
\begin{aligned}
&r = \frac{\ddot{r} - \ddot{r}'}{ch - ch'}, \sigma = \frac{\ddot{\sigma} - \ddot{\sigma}'}{ch - ch'}, \gamma = \frac{\ddot{\gamma} - \ddot{\gamma}'}{ch - ch'}, \theta = \frac{\ddot{\theta_3} - \ddot{\theta_3}'}{ch - ch'}, \\
&d = \frac{\ddot{d} - \ddot{d}'}{ch - ch'}, \beta_1 = \frac{\ddot{\beta_1} - \ddot{\beta_1}'}{ch - ch'}, \beta_2 = \frac{\ddot{\beta_2} - \ddot{\beta_2}'}{ch - ch'}, \gamma = \frac{\ddot{\beta_3} - \ddot{\beta_3}'}{ch - ch'}, \\
&\beta_4 = \frac{\ddot{\beta_4} - \ddot{\beta_4}'}{ch - ch'}, \theta_1 = \frac{\ddot{\theta_1} - \ddot{\theta_1}'}{ch - ch'}, \theta_2 = \frac{\ddot{\theta_2} - \ddot{\theta_2}'}{ch - ch'}, \theta = \frac{\ddot{\theta_3} - \ddot{\theta_3}'}{ch - ch'}.
\end{aligned}
$$

(Special Honest Verifier Zero-knowledge) We construct a simulator $\mathcal{S}$ who is given a random challenge ch. It randomly chooses $\ddot{r}', \ddot{\sigma}', \ddot{\gamma}', \ddot{d}', \ddot{\beta_1}', \ddot{\beta_2}', \ddot{\beta_3}', \ddot{\beta_4}', \ddot{\theta_1}', \ddot{\theta_2}', \dot{\theta_3}', \ddot{\theta_4}' \in Z_{\tilde{p}}^*$ 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, $\dot{u}' = u^{\cdot ch} g^{\ddot{r}'}$ cannot be distinguished from $\dot{u}$ due to randomness.

$$
\begin{aligned}
&\{\dot{u}', \dot{w}', \dot{A}', \dot{C}', \dot{B_{1,1}}', \dot{B_{1,2}}', \dot{B_{3,1}}', \dot{B_{3,2}}', \dot{B_4}', \dot{D}', \ddot{r}', \\
&\ddot{\sigma}', \ddot{\gamma}', \ddot{d}', \ddot{\beta_1}', \ddot{\beta_2}', \ddot{\beta_3}', \ddot{\beta_4}', \ddot{\theta_1}', \ddot{\theta_2}', \dot{\theta_3}', \ddot{\theta_4}'\}.
\end{aligned}
$$

It is indistinguishable from the conversation which is generated by the honest prover.

4.4.2 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 [114]. 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 [142].

• 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 a 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.

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). This happens with negligible probability under the computational Diffie-Hellman assumption. The security proof is similar to the proof of the chosen plaintext security property of the Elgamal encryption. 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

[143]. 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 [144]. 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 [23]. 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 $(r, \sigma, \gamma, a, d)$. This happens with negligible probability under the soundness property of NIZK, 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 [143]. 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 [144]. This happens with negligible probability.

4.5 Performance Evaluation

Since the core component of DAPA is PPIM, we mainly focus on the performance of PPIM. We simulate PPIM and compare our scheme with two existing schemes: VEGS [145] and Vote-to-Link [146], in terms of functionalities and computational costs, and also simulate the communication overhead of PPIM. The existing scheme can only achieve partial functions provided by PPIM, and the comparison results are in Table 4.2. VEGS [145] does not support the dynamic architecture, and a customer cannot be revoked after

the registration. Vote-to-Link [146] does not support the dynamic architecture either, and a customer's identity can only be linked but cannot be recovered or revoked.

Table 4.2: Functionality comparison of PPIM with existing schemes

Properties	Distributed	Dynamic	Confidential	Verifiable	Recoverable	Revocable
PPIM	✓	✓	✓	✓	✓	✓
VEGS [145]	✓	✗	✓	✓	✓	✗
Vote-to-Link [146]	✓	✗	✓	✓	✗	✗

The VEGS is a group-signature-based scheme: a customer can encrypt her group signature and prove to CSSP that the uploaded ciphertext contains a valid group signature that can be verified and opened by VSs (also called the adjudicator in the original paper). Under the circumstances, the encrypted group signatures can be maintained by VSs to guarantee both privacy and accountability. The Vote-to-Link is another distributed identity management scheme that the customer can encrypt her identity credential using a threshold encryption scheme and prove to CSSP that the ciphertext contains a valid credential. In this case, the encrypted credential can be managed by VSs (also called the moderator in the original paper) to guarantee both privacy and accountability. To make the comparison fair, we simplify the calculations of PPIM.

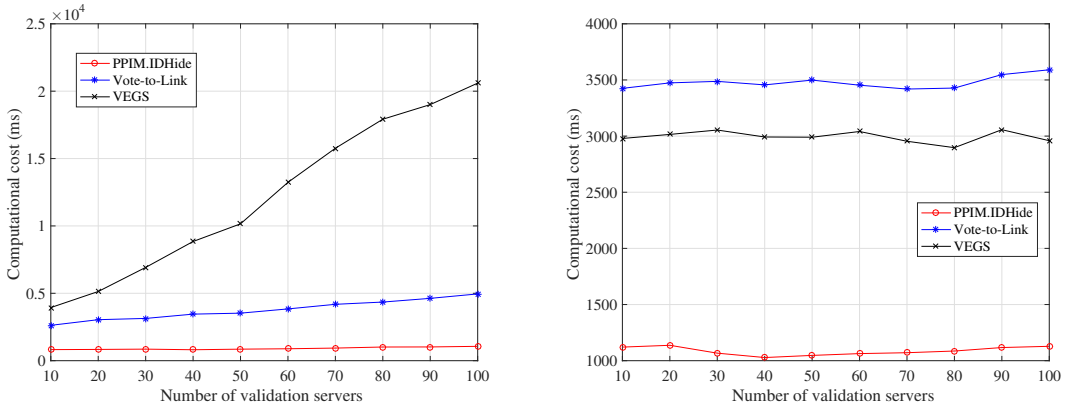
For the cryptographic settings, we utilize the Java pairing based cryptography (JPBC) library [147] and choose the type F pairing (Barreto-Naehrig curve) in our simulation, since it supports asymmetric bilinear pairing. To guarantee security level, the security parameters $\tilde{l}$ is set as $\tilde{l} = 160$.

4.5.1 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 computational complexity, we count the number of time-consuming operations like bilinear pairing and exponentiation in $\tilde{G}$. The computational costs of bilinear pairing is denoted by $PAIR_T$, and the computational costs of exponentiation in $\tilde{G}$ is denoted by $EXP_T^{\tilde{G}}$. 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 (iAn engaged customer means the customer who rents a shared car but does not return it), the results are shown in Table 4.3. *PPIM.IDHide* is the major computational burden for the customer. The customer's computational complexity is $O(N)$ since the customer needs to distribute

the secret $\tilde{y}$ 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 VSs. 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 $O(\chi \cdot N \cdot \hat{t})$ since they need to verify the received χ secrets from N current committee members.

To show the efficiency of PPIM, we also simulate PPIM and two existing schemes VEGS [145] and Vote-to-Link [146]. These schemes are simulated on a Macbook Pro notebook with Intel Core i7 processor. The memory is 8GB and the clock rate is 2.6 GHz. The performance metric used in the comparison is the computational cost of *PPIM.IDHide* with the different number of VSs. The results are shown in Figure 4.5 (a) and (b). PPIM is more efficient than two existing schemes in terms of computational costs and communication overheads. The reason is since 1) the VEGS is designed under the bilinear group of composite order, its computational cost is larger than PPIM which is designed under the bilinear group of prime order; and 2) the Vote-to-Link uses a more time-consuming threshold encryption method and the non-interactive zero-knowledge proof technique has higher computational complexity.



(a) Customer (*PPIM.IDHide*) (b) Validation Server (*PPIM.IDHide*)

Figure 4.5: Computational costs of customers and validation servers in each epoch (*PPIM.IDHide*)

In addition, we also measure the computational cost of *PPIM.IDTransfer* with the different number of VSs. The results are shown in Figure 4.6. To transfer one engaged

customer' secret to the next committee, each current committee member only needs to perform the exponentiation in $\tilde{G}$, which is efficient compared with other complex group operations. Even though the size of next committee is as large as 100, it takes around 400 ms to accomplish the transfer. Similarly, to receive one engaged customer' secret, each next committee member just needs to perform the exponentiation in $\tilde{G}$ as well. Even though the size of current committee is as large as 100, it takes less than $12,000$ 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.

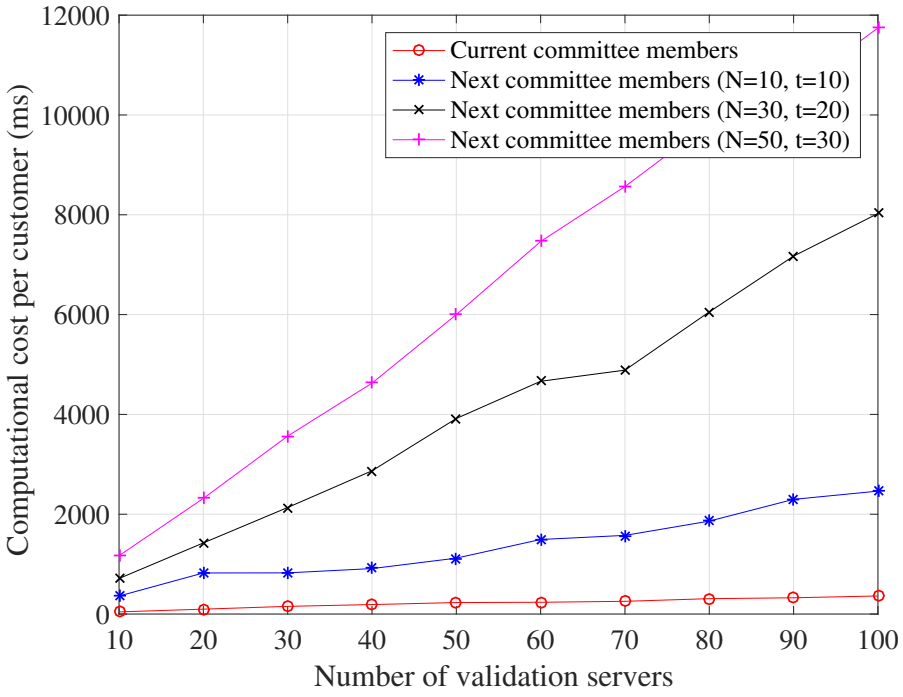


Figure 4.6: Computational costs of customers and validation servers in each epoch (*PPIM.IDTransfer*)

4.5.2 Communication Overheads

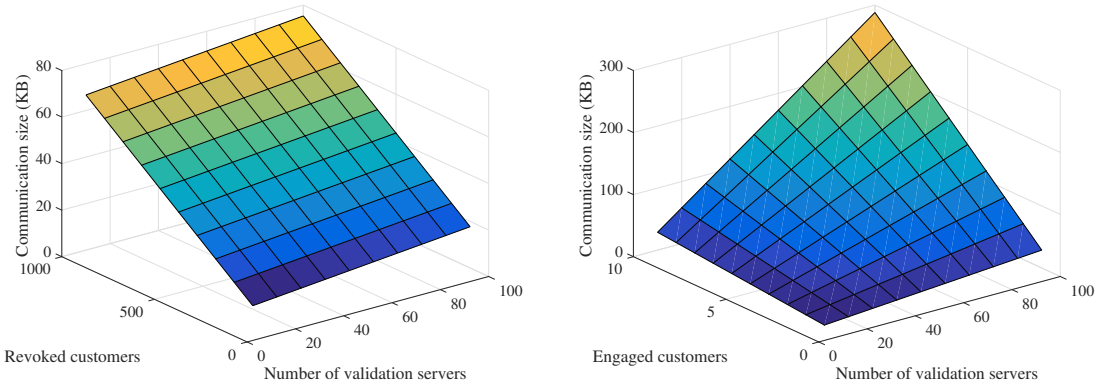
PPIM's communication overheads are mostly related to the number of distributed VSs and the number of engaged customers. If there exist a large number of committee members (VSs), more secrets are needed to be distributed to these VSs when the customer runs *PPIM.IDHide*. When the rental duration is long (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 $Z_{\tilde{p}}$ are denoted by $Size_{\tilde{G}}$, $Size_{G_T}$, and $Size_Z$ the communication overhead of *PPIM.IDHide* between the customer and the current committee member is $(N + t + 14 + n') * Size_{\tilde{G}} + 2 * Size_{G_T} + (N + 12) * Size_Z$. The customer does not only need to upload the public key $\tilde{Y}$ to the bulletin board, share the secret shadow $\{s_i\}_{i=1}^{N}$ with, submit the encrypted secrets (u, w) 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 $\chi((\hat{t} - 1) * Size_{\tilde{G}} + \hat{N} * Size_Z)$. 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 Figure 4.7. Although the communication overheads are linear to the numbers of distributed VSs and engaged customers, it is still acceptable (less than 300 KB).

Table 4.3: Computational complexity of PPIM

PPIM	CSSP	Customer	VS
IDRegister	$EXP_T^{\tilde{G}}$	$3 * PAIR_T + EXP_T^{\tilde{G}}$	-
IDHide	-	$(29 + t + N) * EXP_T^{\tilde{G}} + 6 * PAIR_T$	$(26 + t) * EXP_T^{\tilde{G}} + 9 * Pair_T$
IDTransfer	-	-	$(\chi * \hat{t} + \chi * N * (\hat{t} + 1))EXP_T^{\tilde{G}}$
IDRecover	$(t + 1) * EXP_T^{\tilde{G}}$	-	-
IDRevoke	$EXP_T^{\tilde{G}}$	-	-

4.6 Summary

In this chapter, we have proposed a decentralized, accountable, and privacy-preserving architecture for car sharing services (DAPA). In DAPA, decentralized and dynamic vali-



(a) Between customers and current committee members (*PPIM.IDHide*)

(b) Between current and next committee members (*PPIM.IDTranfer*)

Figure 4.7: Communication overheads of customers and validation servers (epoch)

dation 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.

Chapter 5

Privacy-Preserving Crowdsourcing-based Road Condition Monitoring with Anonymous Reputation Management

5.1 Introduction

With the popularity of smart mobile devices and the advancement of wireless networking technologies, mobile crowdsourcing [148] has become a popular and top-of-the-line approach to achieve real-time data collection and acquisition. Especially in an intelligent transportation system [149], smart vehicles, drivers, and their smartphones can be regarded as mobile sensors to sense and absorb information like real-time road conditions and traffic alerts from the environment. This information can be fed back to drivers and other relative traffic transportation departments to assist in effective traffic management. Compared with traditional road condition monitoring methods based on fixed-deployed sensors, e.g. cameras, the crowdsourcing-based road condition monitoring method offers many advantages, including reducing the costs of sensor deployments, increasing the sensing coverage area, and reducing the reaction delay of local traffic perturbations. For example, WAZE [1] is a typical crowdsourcing-based road condition monitoring system, where users who installed an mobile application named WAZE can register themselves and report different

[1] `https://www.waze.com/`

kinds of road conditions such as traffic jams, road surface harzards, and police traps, to a WAZE monitoring service provider maintained by the Google company. After obtaining enough road condition information, the monitoring service provider can build and update a live road condition map, which can be shared among users and make other vehicle-related services, e.g., navigation, more accurate and convenient.

Although crowdsourcing-based road condition monitoring systems like WAZE bring plenty of benefits, there still exist some realistic security and privacy issues, which may cause privacy concerns and degrade the system's quality. Obviously, the crowdsourcing-based monitoring systems require each user to collect and upload some sensitive information, such as a user's current location to a monitoring service provider, which may violate the privacy of users under privacy laws, such as GDPR [6]. To protect user privacy, some privacy-preserving schemes [90, 91, 89] have been proposed recently. In these schemes, users who care about their privacy generally can conceal their identities (anonymization) or encrypt the road condition information and sensitive data with flexible access controls (encryption) when reporting to a monitoring service provider. These countermeasures can relieve users' privacy concerns and motivate more users to participate in the system, which has been widely acknowledged. Another inevitable risk that the crowdsourcing-based monitoring systems face is data manipulation attacks [150]. We cannot simply assume that all users in the system are honest to submit a real road condition. Large-scale biased and faked road condition reports may lead to serious traffic management issues, e.g., serious traffic jams [151]. To improve the system's quality, not only the academic researchers [98] but also the industrial companies [2] introduce some trust and reputation management mechanisms for users. A user's reputation score is viewed as an important indicator for measuring the reliability of a user's road condition report [152]. In other words, if a user's reputation score is low, his/her report may not appear on the live map. This approach can mitigate the data manipulation attacks, but it somehow conflicts with the privacy requirements of users. Without privacy preservation, any reputation management scheme [153] can be easily integrated with a crowdsourcing-based road condition monitoring system, by enabling a monitoring service provider to bind each user with a life-time reputation score. When receiving a user's report, the monitoring service provider can link the user to a specific reputation score and can easily judge the report's trustworthiness.

However, if the monitoring service provider is not fully trusted, the situation is totally different. A straightforward idea is to deploy a trusted reputation authority in the system to maintain the relationship between a user and his/her reputation score [96] without breaking users' identity privacy. A monitoring service provider can request the trust level of each report from the trusted reputation authority and accordingly update its live map.

[2] `https://support.google.com/waze/partners/answer/6324421?hl=en`

Nevertheless, this design has some drawbacks, e.g., the trusted reputation authority should be always online, suffers from the single point of failure, and cannot be compromised in the real world. To address these weaknesses, another idea is to substitute a centralized trusted authority with decentralized authorities [100]. As long as one authority is honest, adversaries cannot reveal the relationship between a user and his/her reputation score. But this design requires a large number of time-consuming computations and multiple-rounds communications, and decentralized authorities should work together to manage users' reputation scores. Furthermore, some privacy-preserving schemes [97, 98] loose the privacy requirements to improve the efficiency of the reputation management. Instead of achieving high-level user anonymity, they only achieve user pseudonymity. A reputation score is bound with a pseudo-identity of a user, and the user's reports can be linked if he/she uses the same pseudo-identity, which is not desirable since the monitoring service provider could utilize the information included in the report such as locations to deduce some sensitive and private information of the user such as his home address. In this chapter, different from the above-mentioned schemes, we propose a novel privacy-preserving crowdsourcing-based road condition scheme that supports anonymous reputation management without the assistance of any third-party authority. Namely, only users and a monitoring service provider work together to manage users' reputation scores while preserving users' identity privacy. Precisely speaking, even service providers do not know each user's exact reputation score or how it changes in the system, and only users know their exact reputation scores.

There mainly exist two technical challenges in designing such a scheme: 1) how to bind a user's identity credential with his/her reputation score in a privacy-preserving manner; and 2) how to maintain a user's reputation score in a privacy-preserving manner but a monitoring service provider can update the user's reputation score according to his/her reports. The challenge 1) means that a monitoring service provider can authenticate a user's identity credential and his/her bound reputation score included in a road condition reports, but it cannot link the user's two road condition reports. It also implies that a user cannot share his/her reputation score with other users unless the user is willing to share his/her identity credential. The challenge 2) means a monitoring service provider can update a user's reputation score according to the accuracy of the user's reports, without knowing the user's identity or reputation score. To deal with these challenges, we employ a cryptographic building block, i.e., zero-knowledge proof, and other basic cryptographic primitives such as pseudo-random functions, digital signature, and cryptographic homomorphic commitment into our scheme. Specifically, the contributions of this chapter can be summarized as three-folds.

▷ First, we propose a privacy-preserving crowdsourcing-based road condition moni-

toring scheme, where a monitoring service provider can authenticate any registered user's reports and feedbacks but cannot link the user's reports and feedbacks or reveal his/her identity.

- ▷ Second, the proposed scheme additionally achieves anonymous reputation management without the assistance of any online trusted authority or third-party authorities, which is particularly fit for real-world applications. We give a detailed security analysis of the proposed scheme in terms of three security properties: anonymity, K-tolerant trust, and unforgeability.
- ▷ Third, we develop a proof-of-concept prototype based on JAVA, which is utilized to evaluate the performance of the proposed scheme. Based on the experimental results, our scheme is demonstrated to be feasible and practical in terms of computational and communication overhead.

5.2 Models and Design Goals

In this section, we formalize our system model, security model, and identify our design goals.

5.2.1 System Model

In our system model, we consider a privacy-preserving crowdsourcing-based road condition monitoring system, which mainly consists of two entities, namely a monitoring service provider (MSP) and a set of users, as shown in Fig. 5.1.

- ▷ Monitoring Service Provider (MSP): A MSP is a cloud-based service that maintained by an intelligent transportation company, e.g., Google, to crowdsource information from its users (reporter) and offers services on real-time road conditions to its users (receiver). In addition, MSP is also responsible for building a reputation management system for users in a privacy-preserving manner. They can judge the reliability of a road condition report based on the reputation score of the reporter.
- ▷ Users: Users are on-road users, who can register at the MSP and anonymously report to the MSP about the real-time road conditions after the registration, including traffic information, traffic accidents, police traps, and blocked roads. Each user is

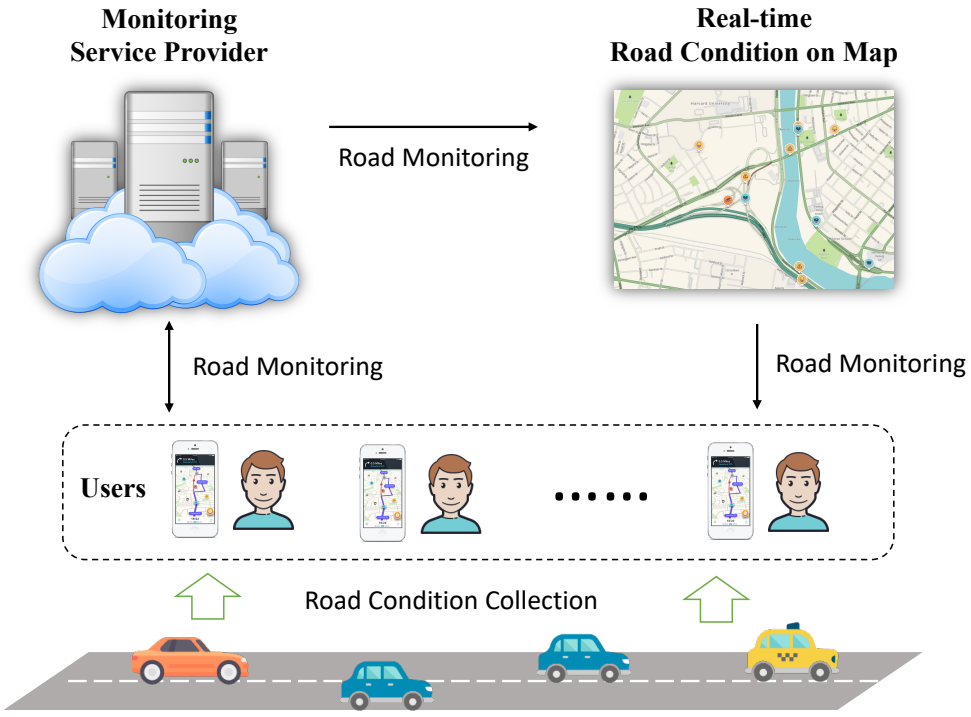


Figure 5.1: System model

assigned a reputation score by the MSP and the initial reputation score for each user is the same. The reputation score can be updated (increased/decreased) by the MSP anonymously, according to the accuracy of his/her reports. The accuracy is determined by the received feedbacks (positive/negative).

Communication Model. Based on traditional certificate-based mechanisms, e.g., one-way SSL/TLS, the communication channel between users and the MSP is assumed to be secure and one-way authenticated (i.e., a user can authenticate a MSP's certificate to verify its identity and a confidential communication channel between them can be established).

5.2.2 Security Model

In our security model, the MSP is assumed to be *honest-but-curious*, i.e., the MSP faithfully follows the system protocols but may be curious about users' identity privacy. This assumption is practically reasonable, as the MSP has to follow the protocols so that the road condition monitoring system can be accepted by users. In the meantime, since some undetectable malwares could have been installed in the MSP to collect users' identity information, we can also assume the MSP is curious and violates users' identity privacy.

On the other hand, we assume the majority of users are honest, but we do not exclude some users may maliciously manipulate their reports and corresponding reputation scores to corrupt the MSP's monitoring service and diminish the accuracy of the road condition monitoring platform. For these reasons, the following three security properties should be satisfied.

- ▷ *Anonymity*: An honest-but-curious MSP cannot reveal the identity of a user who honestly follow the system protocols, and cannot link the user's reports, feedbacks and reputation updating behavior.
- ▷ *K-Tolerant Trust*: A user can only make K reports before updating his/her reputation score, and more than K reports from the same user can be detected by the MSP. In addition, one user can only make one feedback for each report.
- ▷ *Unforgeability*: A user cannot forge his/her identity or modify his/her reputation score binding with his/her identity, i.e., any forged identity and modified reputation score cannot pass the MSP's verification and cannot be recognized as a valid reputation score.

Note that, since this work mainly focuses on privacy preservation, other active attacks, e.g., denial of service (DoS) attack and sybil attack, are beyond the scope of this work, and will be discussed in our future work.

5.2.3 Design Goals

Under the above-mentioned system model and security model, the following three goals should be achieved in the proposed scheme.

- ▷ *Functionality:* The proposed scheme should achieve all basic functions of the system.
- ▷ *Security*: The security properties mentioned in the security model should be realized, including anonymity, K-tolerant trust, and unforgeability.
- ▷ *Feasibility*: The proposed scheme should be feasible to be implemented, including all system protocols and algorithms defined in the scheme.

5.3 Our Proposed Scheme

In this section, we present our privacy-preserving crowdsourcing-based road condition monitoring scheme, which mainly consists of five phases, namely system setup, user registration, data reporting, report feedback, and reputation updating, as shown in Fig. 5.2.

5.3.1 System Setup

Without loss of generality, we consider the MSP bootstraps the whole system in the system setup phase, as the MSP builds the road condition monitoring system by deploying an online road monitoring service and a corresponding mobile application. Any user can download and install the mobile application and utilize the service. The online road minitoring service collects road condition reports from its clients and distills all reports into usable traffic road data to update a real-time map on the client's application. Accordingly, a registered user can freely report to the MSP (i.e., the online road monitoring service) and view the real-time map to obtain the latest road condition information through the mobile application.

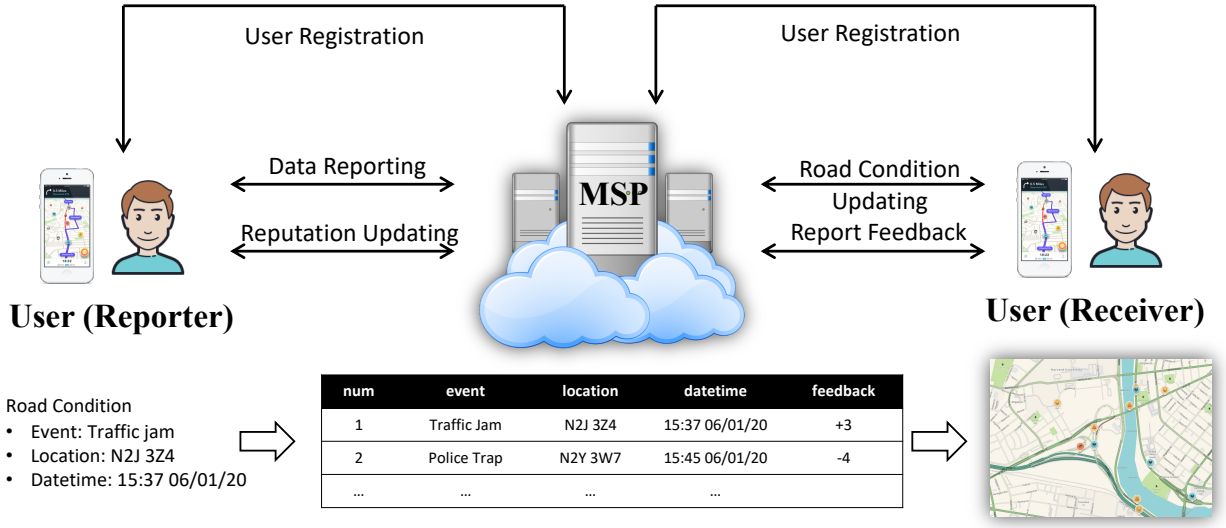


Figure 5.2: Phases of the anonymous crowdsourcing-based road condition monitoring scheme

Specifically, given a security parameter λ, the MSP first chooses three multiplicative cyclic bilinear groups G, $\bar{G}$ and $\mathcal{G}$ of the same prime order q, which satisfies a bilinear map $e : G \times \bar{G} \rightarrow \mathcal{G}$. The length of the prime order q is λ, and there does not exist an efficiently computable isomorphism $\phi : \bar{G} \rightarrow G$, i.e., e is called a type-3 pairing. Then, the MSP selects 3 independent generators $(\mathfrak{g}, g, h)$ of G^3 and a generator $\bar{g}$ of $\bar{G}$. Moreover, the MSP sets a range $[0, 2^{\kappa}]$ for reputation scores, and chooses two cryptographic hash functions $H : \{0,1\}^* \rightarrow Z_q$ and $H' : \{0,1\}^* \rightarrow G$. Also, the MSP chooses empty sets UList, IList, SList and sets an accumulated value $accu = \mathfrak{g}$ for the purpose of identity management. Finally, the MSP publishes the public parameters pp, where

$$pp = (\lambda, q, g, h, \bar{g}, G, \bar{G}, \mathcal{G}, e, H, H', accu, \mathsf{IList}, \mathsf{SList}).$$

Next, based on the public parameters pp, MSP generates a pair of public and private keys (pk, sk) as follows and publishes the public key pk: i) Set an upper bound n for the accumulated value $accu$; ii) Choose 5 random elements $(x, y, z, v, w, u) \in_R Z_q^5$; iii) Generate the private key

$$sk = (X = g^x, u)$$

; and iv) Generate the corresponding public key

$$pk = (Y, Z, V, W, \bar{X}, \bar{Y}, \bar{Z}, \bar{V}, \bar{W}, \mathbf{U}, \bar{U})$$

Algorithm 1 URegister Algorithm

Input: (pp, pk).
Output: $(uid, r, id, sn, ID, \pi_{ID})$.
1: Choose a random element $r \in_R Z_q$.
2: Choose a pseudo identity $id \in_R Z_q$.
3: Choose a unique identity $uid \in_R Z^q$.
4: Choose a serial number $sn \in_R Z_q^K$.
5: Generate a blinded pseudo identity $ID = g^r W^{sn} Y^{id} Z^{uid}$.
6: Generate a non-interactive zero-knowledge proof $\pi_{ID} = \text{NIZK}_1\{(r, id, sn, uid) : ID = g^r W^{sn} Y^{id} Z^{uid}\}$.
7: Output $(uid, r, id, sn, ID, \pi_{ID})$.

where $Y = g^y$, $Z = g^z$ $V = g^v$, $W = g^w$, $\mathbf{U} = \{U_1 = \mathfrak{g}^u, U_2 = \mathfrak{g}^{u^2}, ..., U_n = \mathfrak{g}^{u^n}\}$, $\bar{X} = \bar{g}^x$, $\bar{Y} = \bar{g}^y$, $\bar{Z} = \bar{g}^z$, $\bar{V} = \bar{g}^v$, $\bar{W} = \bar{g}^w$, and $\bar{U} = \bar{g}^u$.

5.3.2 User Registration

In order to participate in the crowdsourcing-based road condition monitoring system, each user first needs to real-name authenticate himself/herself to the MSP. Note that, if a user has registered himself/herself before, the user cannot register again. After the real-name authentication, the user runs an algorithm URegister as shown in Algorithm 1 to generate the registration information $(uid, r, id, sn, ID, \pi_{ID})$, stores (uid, r, id, sn) locally, sends the registration request (ID, π_{ID}) to the MSP, and waits for the response.

After receiving the registration request from the user, the MSP verifies the zero-knowledge proof π_{ID} and responds FAIL back to the user if the proof is not correct. If the verification passes, the MSP chooses an initial reputation score $\theta \in [0, 2^\kappa]$ for the user, and generates a signed blinded pseudo identity $SID = (SID_1 = g^{\hat{r}}, SID_2 = (X \cdot V^\theta \cdot ID)^{\hat{r}})$. Finally, the MSP responds (SUCCESS, SID, θ, $accu$, SList) back to the user, where $accu$ is the current accumulated value and SList is the latest set used for storing the used serial numbers.

After receiving the response from the MSP, the user checks the response. If the response is FAIL, the user fails to make the registration. If the response is SUCCESS, the user can run an algorithm UCredGen as shown in Algorithm 2 to generate a signed pseudo identity PID and a non-membership witness wit.

Algorithm 2 UCredGen Algorithm

Input: $(pp, pk, r, SID, \theta, id, sn, \mathsf{SList})$.
Output: FAIL or PID.
1: Set $PID_1 = SID_1 = \sigma$.
2: Generate $PID_2 = SID_2 \cdot \sigma^{-r}$.
3: Set $PID = (PID_1, PID_2)$.
4: Verify $e(PID_1, \bar{X}\bar{V}^{\theta}\bar{Y}^{id}\bar{W}^{sn}\bar{Z}^{uid}) = e(PID_2, \bar{g})$.
5: **if** the verification does not pass **then**
6: Output FAIL.
7: **end if**
8: Generate a polynomial $f(\rho) = \prod_{i=1}^{n'} c_i \rho^i$ satisfies $\prod_{sn' \in \mathsf{SList}}(sn' + \rho) = f(\rho) \cdot (sn + \rho) + d$, where c_i is the coefficient of the polynomial $f(\rho)$, n' is the size of SList, and d is a constant.
9: Generate a non-membership witness $wit = (wit_1, wit_2)$, where $wit_1 = \mathfrak{g}^{f(c_i)} = \prod_{i=1}^{n'} U_i^{c_i}$ and $wit_2 = d$.
10: Output (PID, wit).

Finally, if the UCredGen algorithm does not output FAIL, the user locally stores

$$Cred = (uid, id, sn, PID, wit, \theta)$$

as an anonymous credential and the registration completes.

Note that, when an additional sn' is added into SList, the user will be notified and can update his/her non-membership witness $wit = (wit_1, wit_2)$ by the following equations.

$$wit_1 = accu \cdot wit_1^{sn'-sn}, \; wit_2 = wit_2 \cdot (sn' - sn).$$

If n' almost reaches n, where n' is current size of SList, the MSP needs to update its public key pk and private key sk, notifies the users to update their anonymous credential $Cred$, and re-set the set SList to be empty. The procedure of anonymous credential updating is similar to the procedure of the reputation updating, which will be given in the following.

5.3.3 Data Reporting

A registered user can anonymously make a report (M, RID, tk, π_M, T) to the MSP, including the details of a road condition M, a randomized pseudo identity RID, a one-time token tk, a corresponding signature proof of knowledge π_M, and a tag T for marking the

Algorithm 3 UReport Algorithm

Input: $(pp, pk, Cred)$.

Output: FAIL or $(M, RID, tk, \bar{\theta}, \pi_M, T)$.

if $sn \in$ SList **then**
  Output FAIL.
end if
repeat
  Choose $k \in [0, K-1]$.
  Calculate $tk = g^{\frac{1}{id+k+1}}$
until $tk \notin$ IList
Choose two random elements $(\beta, \gamma) \in_R Z_q^2$.
Set a randomized reputation score $\bar{\theta} \leq \theta$.
Set a road condition M.
Generate a unique tag $T = H'(M)^{uid}$.
Randomize the signed pseudo identity PID as $RID = (RID_1, RID_2)$ where $RID_1 = PID_1^{\beta}$ and $RID_2 = (PID_2 \cdot (PID_1)^{\gamma})^{\beta}$.
Generate a non-interactive signature proof of knowledge π_M as follows.

$$\begin{aligned}
&\pi_M = \mathrm{NIZK}_2\{(\theta, \gamma, id, sn, wit_1, wit_2, uid, k) : \\
&\wedge e(RID_2, \bar{g}) = e(RID_1, \bar{X}\bar{V}^{\theta}\bar{W}^{sn}\bar{Y}^{id}\bar{Z}^{uid}) \\
&\cdot e(RID_1, \bar{g})^{\gamma} \wedge e(wit_1, \bar{g}^{sn} \cdot \bar{U}) \cdot e(\mathfrak{g}^{wit_2}, \bar{g}) \\
&= e(accu, \bar{g}) \wedge wit_2 \neq 0 \wedge T = H'(M)^{uid} \wedge \\
&tk = g^{\frac{1}{id+k+1}} \wedge 0 \leq k \leq K-1 \wedge \bar{\theta} \leq \theta \leq 2^{\kappa} - 1\}(M)
\end{aligned}$$

Output $(M, RID, tk, \bar{\theta}, \pi_M, T)$.

reporter. The road condition M consists of one of many types of information (e.g., map issues, gas prices, traffic jams, and road closures, etc), the timestamp, and the location of road condition. Specifically, the user runs an algorithm $\mathsf{UReport}$ as shown in Algorithm 3 to generate the report $(M, RID, tk, \bar{\theta}, \pi_M, T)$, sends the report to the MSP, and waits for the response. If the algorithm returns FAIL, it means the anonymous credential is not valid, since the serial number sn has been used.

After receiving a report, the MSP verifies the correctness of the report by the following steps. First, it checks whether the one-time token tk exists in IList. If $tk \in \mathsf{IList}$, the MSP rejects the report and responds REJECT back to the user since the report uses an old token. Otherwise, the MSP adds the token tk into IList. Then, the MSP verifies the signature proof of knowledge π_M. If the verification does not pass, the MSP rejects the report and responds REJECT back to the user since the zero-knowledge proof is not correct. Otherwise, it marks the report with a unique report number num, marks the reporter $O = T$, creates an empty set FList for the report's feedback management, adds T into FList, initializes a feedback index $B = 0$, stores $(num, O, M, \mathsf{FList}, B)$ into its local database, and responds (ACCEPT, num) back to the user.

After receiving the response from the MSP, the user acknowledges that his/her report has been accepted or rejected, locally stores num, and the data reporting phase completes. At the same time, if the report is successful, the MSP updates the real-time map on its online road monitoring service. The updated map is synchronized on the mobile application at the client side, and other users can view the updated map and the report on their mobile applications.

5.3.4 Report Feedback

When a road condition is updated on the real-time map, other users who locate in a near region can view it. Also, they can give a positive or negative feedback to the report through tapping on a thumbs-up symbol or a thumbs-down symbol on their mobile application, indicating that the report is accurate or not. One user, except the user who made the report, can give one and only one feedback to the report. In particular, the user runs an algorithm UFeed as shown in Algorithm 4 to generate the feedback information $(\Delta, RID, \pi_\Delta, T)$ according to the content of the report M, sends the feedback $(num, \Delta, RID, \pi_\Delta, T)$ to the MSP, and waits for the response.

After the MSP receives the feedback, it can locate the feedback set FList and the feedback index B according to the report number num, and checks whether T exists in $FList$. If $T \in \mathsf{FList}$, the MSP responds REJECT back to the user since it is a repeated

Algorithm 4 UFeed Algorithm

Input: $(pp, pk, Cred, M)$.

Output: $(\Delta, RID, \pi_\Delta, T)$.

1: Choose two random elements $(\epsilon, \eta) \in_R Z_q^2$.

2: Generate a unique tag $T = H'(M)^{uid}$.

3: Randomize the signed pseudo identity PID as $RID = (RID_1, RID_2)$ where $RID_1 = PID_1^\eta$ and $RID_2 = (PID_2 \cdot (PID_1)^\epsilon)^\eta$.

4: Set a feedback $\Delta = (num, 1/0)$, where 1 denotes a positive feedback and 0 denotes a negative feedback.

4: Generate a non-interactive signature proof of knowledge π_Δ as follows.

$$\pi_\Delta = \text{NIZK}_3\{(\epsilon, \theta, id, sn, uid) :$$
$$e(RID_2, \bar{g}) = e(RID_1, \bar{X}\bar{V}^\theta \bar{W}^{sn} \bar{Y}^{id} \bar{Z}^{uid})$$
$$\cdot e(RID_1, \bar{g})^\epsilon \wedge T = H'(M)^{uid}\}(\Delta)$$

4: Output $(\Delta, RID, \pi_\Delta, T)$.

feedback. If $T \notin$ FList, the MSP verifies the signature proof of knowledge π_Δ. If the verification does not pass, the MSP responds REJECT back to the user since the zero-knowledge proof is not correct. Otherwise, the MSP accepts the feedback. If the feedback is positive, the feedback index of the report B increases ($B = B + 1$), and the report will last longer on the real-time map. Otherwise, B decreases ($B = B - 1$) and the report will disappear very soon. The MSP finally responds ACCEPT back to the user. After receiving the response, the user acknowledges that his/her feedback is accepted or rejected, and the report feedback phase completes.

5.3.5 Reputation Updating

Assuming that there exist a report accuracy measurement method based on the feedback index B, the MSP can judge the accuracy of a report and update the corresponding user's reputation score who made the report. For example, if B is larger than a threshold, the MSP believes that the report is accurate. Concretely, according to the report number num and the content of the report M, a registered user who previously reported the road condition can runs an algorithm UUpdate as shown in Algorithm 5 to generate reputation updating information $(\bar{r}, \overline{id}, \overline{sn}, \overline{ID}, RID, T, \pi_{RP})$, locally stores $(\bar{r}, \overline{id}, \overline{sn})$, and sends the reputation updating request $(num, T, \overline{ID}, RID, sn, \pi_{RP})$ to the MSP.

Algorithm 5 UUpdate Algorithm

Input: $(pp, pk, num, M, Cred)$.

Output: $(\bar{r}, \overline{id}, \overline{sn}, \overline{ID}, T, RID, \pi_{RP})$.

1: Choose a random element $\bar{r} \in_R Z_q$.

2: Choose two random elements $(\omega, \tau) \in_R Z_q^2$.

3: Choose a new pseudo identity $\overline{id} \in_R Z_q$.

4: Choose a new serial number $\overline{sn} \in_R Z_q$.

5: Set an updated blinded pseudo identity $\overline{ID} = g^{\bar{r}} V^{\theta} W^{\overline{sn}} Y^{\overline{id}} Z^{uid}$.

6: Randomize the signed pseudo identity PID as $RID = (RID_1, RID_2)$ where $RID_1 = RID_1^{\omega}$ and $RID_2 = (PID_2 \cdot (PID_1)^{\tau})^{\omega}$.

7: Generate a unique tag $T = H'(M)^{uid}$.

8: Generate a non-interactive zero-knowledge proof π_{RP} as follows.

$$\begin{aligned}\pi_{RP} = \text{NIZK}_4\{&(\bar{r}, \tau, \overline{id}, \overline{sn}, \theta, id, uid) : T = H'(M)^{uid} \\ &\wedge \overline{ID} = g^{\bar{r}} V^{\theta} W^{\overline{sn}} Y^{\overline{id}} Z^{uid} \wedge e(RID_2, \bar{g}) \\ &= e(RID_1, \bar{X}\bar{V}^{\theta}\bar{W}^{sn}\bar{Y}^{id}\bar{Z}^{uid}) \cdot e(RID_1, \bar{g})^{\tau}\}\end{aligned}$$

9: Output $(\bar{r}, \overline{id}, \overline{sn}, \overline{ID}, T, RID, \pi_{RP})$.

After receiving the reputation updating request, the MSP first locates the report $(num, O, M, \mathsf{FList}, B)$ according to the report number num, checks whether $O = T$ and sn exist in SList. If $O \neq T$, the MSP responds FAIL back to the user since the the user is not owner of the report. If $sn \in \mathsf{SList}$, the MSP responds FAIL back to the user since the user's pseudo identity is not valid. Then, the MSP verifies the zero-knowledge proof π_{RP} and responds FAIL back to the user if the zero-knowledge proof is not correct. Otherwise, according to the accuracy of the report, the MSP sets a reputation change θ', e.g., $\theta' = 1$ or $\theta' = -1$. Next, the MSP chooses a random element $l \in_R Z_q$, and generate a signed blinded pseudo identity $SID = (SID_1, SID_2)$, where $SID_1 = g^l$ and $SID_2 = (X \cdot V^{\theta'} \cdot \overline{ID})^l)$. Also, the MSP updates the accumulated value $accu = accu^{u+sn}$, adds sn into SList, and sends the reputation updating response (SUCCESS, SID, θ', $accu$, SList) back to the user.

After receiving the response from the MSP, the user checks the response. If the response is FAIL, the user fails to complete the reputation updating. If the response is SUCCESS, the user first updates his/her reputation as $\theta = \theta + \theta'$ and updates his/her pseudo identity $id = \overline{id}$ and serial number $sn = \overline{sn}$. Then, the user can run the algorithm $\mathsf{UCredGen}$ as shown in Algorithm 2 by inputing $(pp, pk, \bar{r}, SID, \theta, id, sn, \mathsf{SList})$ to generate a new signed pseudo identity PID and a non-membership witness wit. Finally, the user locally stores

$$Cred = (uid, id, sn, PID, wit, \theta)$$

as an updated anonymous credential and the reputation updating completes.

5.4 Security Analysis

In this section, we analyze the security requirements defined in the security model, and prove that the proposed scheme achieves anonymity, K-tolerant trust, and unforgeability. Since the zero-knowledge proof is one of the most significant constructions in our scheme, before showing the detailed security analysis, we first demonstrate that the proposed zero-knowledge proofs achieves completeness, soundness, and zero-knowledge.

Theorem-1: The zero-knowledge proofs ($NIZK_1$, $NIZK_2$, $NIZK_3$, $NIZK_4$) in the proposed scheme achieves completeness, soundness, and zero-knowledge properties.

Proof: A standard $\Sigma-$protocol [154] is one of the most popular approaches to instantiate a zero-knowledge proof, which can achieve completeness, special soundness, and honest-verifier zero-knowledge. Completeness denotes that on input a public input x and a private witness w where $(x, w) \in R$ (R is a non-deterministic polynomial-time relation), a prover $\mathcal{P}$ can prove to a verifier $\mathcal{V}$ that $(x, w) \in R$, and $\mathcal{V}$ always accepts $\mathcal{P}$'s proof. Special

soundness denotes that there always exists an extractor that can efficiently extract w from the the proof which satisfies $(x, w) \in R$, if receiving any x and any pair of accepting conversations on input x with different random challenges ch and ch' ($ch \neq ch'$). Honest-verifier zero-knowledge denotes that there always exists a polynomial-time simulator $\mathcal{S}$ with the input x and a random challenge ch, that can output an accepting conversation, with the same probability distribution as conversations between a honest prover and a honest verifier. In a random oracle model, an interactive $\Sigma-$ protocol [154] can be easily transferred to a non-interactive zero-knowledge proof NIZK that satisfies completeness, soundness, and zero-knowledge properties based on Fiat-Shamir Heuristic [154].

As we discussed above, since NIZK_1, NIZK_3, and NIZK_4 are non-interactive proofs of discrete logarithm relation, which can be instantiated through a standard $\Sigma-$protocol [154], NIZK_1, NIZK_3, and NIZK_4 achieve completeness, soundness, and zero-knowledge. NIZK_2 can be formed through several sub-proofs (NIZK_5, NIZK_6, NIZK_7) as follows. Additional three random elements $(\alpha, \alpha', \bar{\alpha}) \in_R Z_q^3$ and commitments com, com', and $\overline{com}$ are introduced to link these sub-proofs for easy understanding.

$$
\begin{aligned}
&\text{NIZK}_5\{(\theta, \alpha, \alpha', \bar{\alpha}, id, sn, x, uid) : \\
&com = g^{\theta} h^{\alpha} \wedge com' = g^{sn} h^{\alpha'} \wedge \overline{com} = g^{x} h^{\bar{\alpha}} \wedge \\
&e(RID_2, \bar{g}) = e(RID_1, \bar{X}\bar{V}^{\theta}\bar{W}^{sn}\bar{Y}^{id}\bar{Z}^{uid}) \cdot e(RID_1, \bar{g})^{\gamma} \\
&\wedge T = H'(M)^{uid} \wedge e(tk, \bar{g}^{id}\bar{g}^{x}\bar{g}) = e(g, \bar{g})\}
\end{aligned}
$$

NIZK_5 is a proof of discrete logarithm relation. Similar to NIZK_1, NIZK_3, and NIZK_4, NIZK_5 achieves completeness, soundness, and zero-knowledge.

$$
\begin{aligned}
&\text{NIZK}_6\{(sn, \alpha', wit_1, wit_2) : com' = g^{sn} h^{\alpha'} \wedge wit_2 \neq 0 \\
&\wedge e(wit_1, \bar{g}^{sn} \cdot \bar{U}) \cdot e(\mathfrak{g}^{wit_2}, \bar{g}) = e(accu, \bar{g})\}
\end{aligned}
$$

NIZK_6 is a non-interactive proof of knowledge of a committed element not in an accumulator value, which has been clearly defined in [33] and has been proven to be complete, sound, and zero-knowledge.

$$
\begin{aligned}
&\text{NIZK}_7\{(x, \theta, \alpha, \bar{\alpha}) : \overline{com} = g^{x} h^{\bar{\alpha}} \wedge x \in [0, K-1] \\
&\wedge com = g^{\theta} h^{\alpha} \wedge \theta \in [\bar{\theta}, 2^{\kappa} - 1]\}
\end{aligned}
$$

NIZK_7 is a non-interactive zero-knowledge range proof, which can handle an arbitrary range through the method proposed in [155] and can be easily transformed to a standard

range proof. The proof has been proven to be complete, sound, and zero-knowledge in [156]. Combining NIZK_5, NIZK_6, and NIZK_7, we can easily derive the non-interactive zero-knowledge proof NIZK_2, and thus NIZK_2 achieves completeness, soundness, and zero-knowledge.

Theorem-2: The proposed scheme achieves anonymity.

Proof: Let $\mathcal{A}$ be a probabilistic polynomial-time adversary who controls the MSP. To prove that the proposed scheme achieves anonymity, we can prove that $\mathcal{A}$ cannot distinguish the output of a honest user from the output of a simulator $\mathcal{S}$ after engaging in a legal number of data reporting protocols, report feedback protocols, and reputation updating protocols.

For each data reporting procedure, $\mathcal{S}$ can simulate the output (M, RID, tk, π_M, T) as follows.

- $\triangleright$ $\mathcal{S}$ sets a report condition M.
- $\triangleright$ $\mathcal{S}$ chooses a randomness $uid \in_R Z_q$ and compute $T = H'(M)^{uid}$.
- $\triangleright$ $\mathcal{S}$ chooses a randomness $id \in_R Z_q$ and a randomness $k \in_R [0, 1, ..., K-1]$, and compute $tk = g^{\frac{1}{id+k+1}}$.
- $\triangleright$ $\mathcal{S}$ chooses two random group elements $RID = (RID_1, RID_2) \in_R G^2$.
- $\triangleright$ $\mathcal{S}$ simulates a zero-knowledge proof π_M of θ, γ, id, sn, wit_1, wit_2, uid, k such that
  1. $e(RID_2, \bar{g}) = e(RID_1, \bar{X}\bar{V}^{\theta}\bar{W}^{sn}\bar{Y}^{id}\bar{Z}^{uid}) \cdot e(RID_1, \bar{g})^{\gamma}$
  2. $e(wit_1, \bar{g}^{sn} \cdot \bar{U}) \cdot e(\mathfrak{g}^{wit_2}, \bar{g}) = e(accu, \bar{g}) \wedge wit_2 \neq 0$
  3. $T = H'(M)^{uid} \wedge tk = g^{\frac{1}{id+k+1}}$
  4. $0 \leq k \leq K-1 \wedge \bar{\theta} \leq \theta \leq 2^{\kappa} - 1$

The output of $\mathcal{S}$ is computationally indistinguishable from the output of an honest user due to the following reasons. According to the user registration protocol, $\mathcal{A}$ learns nothing about the secrets (uid, id) since the commitment ID is perfect hiding [157] and the proof π_{ID} is zero-knowledge. Hence, the values (uid, id) chosen by $\mathcal{S}$ are distinguishable from those chosen by an honest user. Due to the security of the pseudo-random function, tk is distinguishable from a random element in the group G under the q-DDHI assumption [30]. Considering that H' is a random oracle, T is distinguishable from a random element in the group G. RID_1 and RID_2 are distinguishable from two random elements in the

group G due to the security of a PS signature [26]. The proof π_M can be simulated by $\mathcal{S}$ and the simulated proof is indistinguishable from a real proof since the proof is proven to be zero-knowledge. Therefore, the probability that $\mathcal{A}$ can distinguish a real user and a simulator $\mathcal{S}$ is negligible. Similarly, for each report feedback procedure and each reputation updating procedure, $\mathcal{S}$ can simulate the output $(num, \Delta, RID, \pi_\Delta, T)$ and the output $(num, T, \overline{ID}, RID, sn, \pi_{RP})$. In the random oracle model, $\mathcal{A}$ can distinguish a real user and a simulator $\mathcal{S}$ if and only if it could distinguish real proofs or simulated proofs, or it could break the security of the PS signature, or it could break the perfect hiding property of a Pedersen commitment. However, the probability is negligible.

Finally, we can conclude that the proposed scheme achieves anonymity.

Theorem-3: The proposed scheme achieves K-tolerant trust.

Proof: Let $\mathcal{A}$ be a probabilistic polynomial-time adversary who executes the user registration protocol, data reporting protocol, report feedback protocol, and reputation updating protocol with a simulator $\mathcal{S}$ acting as an honest MSP. For each user registration request, $\mathcal{S}$ behaves exactly like an honest MSP but additionally utilizes the extractor of the zero-knowledge proof NIZK_1 to extract the secrets (id, sn, uid). For each reputation updating request, $\mathcal{S}$ behaves exactly like an honest MSP but additionally utilizes the extractor of the zero-knowledge proof NIZK_4 to extract the secrets (id, sn). Based on the value id, $\mathcal{S}$ could calculate a tuple $\vec{\chi} = (\chi_1, \chi_2, ..., \chi_K)$ where $\chi_i = g^{\frac{1}{id+i}}$ for $i = 1$ to K. Based on the value uid, $\mathcal{S}$ could calculate a tag $T' = H'(M)^{uid}$ according to the message M. In addition to black-box access to $\mathcal{A}$, $\mathcal{S}$ also controls over the random oracles H and H'. Due to the soundness of the zero-knowledge proof, $\vec{\chi}$ contains all valid one-time tokens that $\mathcal{A}$ can generate, except with negligible probability.

In order to break K-tolerant trust, one of the following three cases happens: i) $\mathcal{A}$ convinces an honest MSP to accept an invalid one-time token tk and he/she can generate a valid zero-knowledge proof for the token with some non-negligible probability; ii) $\mathcal{A}$ convinces an honest MSP to accept an invalid serial number sn and he/she can generate a valid zero-knowledge proof for the serial number with some non-negligible probability; and ii) $\mathcal{A}$ convinces an honest MSP to accept an invalid tag T and can generate a valid zero-knowledge proof for the tag with some non-negligible probability.

For the case i) where $\mathcal{A}$ convinces an honest MSP to accept an invalid one-time token tk during the data reporting protocol, $\mathcal{A}$ must conduct a proof such that one of the following statements is fake: 1) $e(RID_2, \bar{g}) = e(RID_1, \bar{X}\bar{V}^{\theta}\bar{W}^{sn}\bar{Y}^{id}\bar{Z}^{uid}) \cdot e(RID_1, \bar{g})^{\gamma}$; 2) $tk = g^{\frac{1}{id+k+1}}$; and 3) $0 \leq k \leq K-1$. The statement 1) can be faked with negligible probability under the PS assumption, as violating statement 1) implies breaking the unforgeability of the PS signature [26]. The statement 2) can be faked with negligible probability under

the q-DDHI assumption [30] and the statement 3) can be faked with negligible probability under the DL assumption [156]. To conclude, the total success probability of $\mathcal{A}$ in case i) is negligible. Therefore, $\mathcal{A}$ must use a valid one-time token tk to achieve one data reporting. To report more than K times, $\mathcal{A}$ must use duplicated tokens with overwhelming probability, and an honest MSP can detect it by checking the set IList. Otherwise, $\mathcal{A}$ must update his/her pseudo-identity id, serial number sn, and reputation θ by running the reputation updating protocol with the honest MSP.

For the case ii) where $\mathcal{A}$ convinces an honest MSP to accept an invalid serial number sn during the data reporting protocol, $\mathcal{A}$ must conduct a proof such that the statement 1) defined in case i) is faked and the statement $e(wit_1, \bar{g}^{sn} \cdot \bar{U}) \cdot e(\mathfrak{g}^{wit_2}, \bar{g}) = e(accu, \bar{g}) \wedge wit_2 \neq 0$ is faked. The statement can be faked with negligible probability under the q-SDH assumption [33], as violating the statement implies breaking the security of the accumulator [33]. To conclude, the total success probability of $\mathcal{A}$ in case ii) is negligible. Therefore, $\mathcal{A}$ must use a valid serial number sn to achieve one data reporting. If $\mathcal{A}$ uses old serial numbers to make requests, an honest MSP can detect it by checking the set SList and the requests are not accepted.

For the case iii) where $\mathcal{A}$ convinces an honest MSP to accept an invalid tag T during the report feedback protocol, $\mathcal{A}$ must conduct a proof such that the statement 1) defined in case i) is faked and the statement $T = H'(M)^{uid}$ is fake. The statement can be faked with negligible probability under the DL assumption. To conclude, the total success probability of $\mathcal{A}$ in case iii) is negligible. Therefore, $\mathcal{A}$ must use a valid tag T to make the feedback for a specific report M. When a user makes more than one feedbacks for the same report M, an honest MSP can detect it by checking the set FList.

Since all three cases happen with negligible probability, the proposed scheme achieves K-tolerant trust.

Theorem-4: The proposed scheme achieves unforgeability.

Proof: The unforgebaility of an anonymous credential $Cred$ is quite straightforward due to the unforgeability of the PS signature, i.e., the probability that an probabilistic polynomial-time adversary $\mathcal{A}$ can forge a valid anonymous credential is negligible under the PS assumption [26]. In addition, if $\mathcal{A}$ wants to convince an honest MSP to accept an forged/modified reputation score θ during the data reporting protocol, $\mathcal{A}$ must conduct a proof such that one of the following statements is fake: 1) $e(RID_2, \bar{g}) = e(RID_1, \bar{X}\bar{V}^{\theta}\bar{W}^{sn}\bar{Y}^{id}\bar{Z}^{uid}) \cdot e(RID_1, \bar{g})^{\gamma}$; and 2) $\theta' \leq \theta \leq 2^{\kappa} - 1$. As we discussed above, the statement 1) and 2) can be faked with negligible probability. As a result, the total success probability of $\mathcal{A}$ is negligible and the proposed scheme achieves unforgeability.

5.5 Performance Evaluation & Implementation

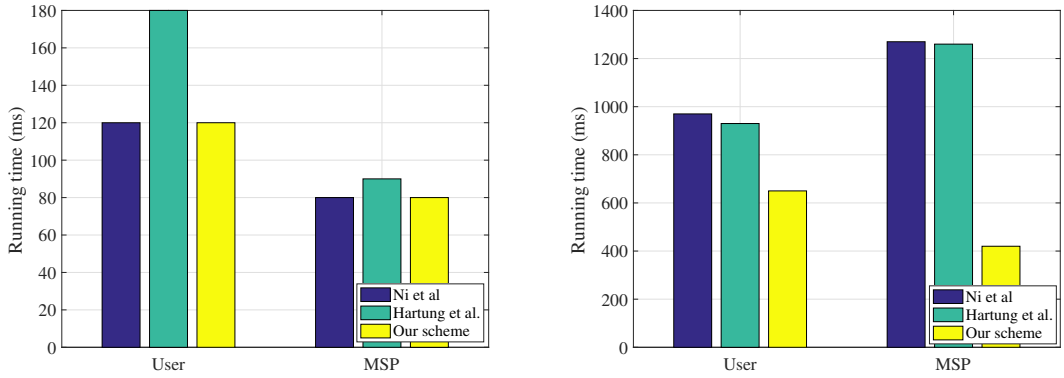
In this section, we evaluate the performance of the proposed scheme in terms of computational and communication overhead. The performance evaluation involves two parts: 1) we analyze the computational and communication complexity of our scheme and two existing schemes [102, 103]; and 2) we implement our scheme on a Macbook laptop and an Android smartphone to demonstrate its feasibility and efficiency.

5.5.1 Complexity Analysis

We compare our scheme with Ni et al.'s scheme [102] and Hartung et al.'s scheme [103] in terms of computational and communication complexity. The reason that we choose these schemes is that they also address the identity privacy issue and the reputation management issue at the same time without the assistance of online third parties. In short, Ni et al.'s scheme [102] is designed based on a BBS+ group signature scheme [30] and a zero-knowledge range proof scheme [155], and Hartung et al.'s scheme [103] is designed based on a structure-preserving signature scheme [158] and a zero-knowledge range proof scheme [155]. However, these schemes cannot be directly applied in our scenario, since they do not support K-tolerant trust property. Therefore, we do not consider the K-tolerant trust property in the comparison and we simplify our scheme to make the comparison fair.

For computational complexity, we count the number of two time-consuming operations in these schemes: exponentiation operations in G ($\mathcal{E}$) and bilinear pairing operations ($\mathcal{P}$). Other operations such as hashing operations and multiplication operations are neglected as they can be done in ignorable time compared with $\mathcal{E}$ and $\mathcal{P}$. The comparison results are shown in Table 5.1. Compared with Ni et al.'s scheme [102] and Hartung et al.'s scheme [103], our scheme is more computationally efficient since our scheme is constructed based on a more efficient PS signature [26] and a more efficient zero-knowledge range proof scheme [156]. For clearly showing the comparison of computational overhead, we also give numeric results of a simulation in Figure 5.3, assuming that the reputation score range is $[0, 1023]$ (i.e., $\kappa = 10$), the running time of an exponentiation operation in G is 10 ms, the running time of a bilinear pairing operation is 20 ms (Note that, the running time is an approximate value which may be varied using different programming languages on different testbeds).

For communication complexity, we count the number of elements in Z_p, G and $\mathcal{G}$ transmitted between users and MSPs in these schemes. The sizes of Z_p, G and $\mathcal{G}$ are denoted by $|Z_p|$, $|G|$ and $|\mathcal{G}|$, and the comparison results are shown in Table 5.2, and our



(a) User Registration

(b) Data Reporting

(c) Report Feedback

(d) Reputation Updating

Figure 5.3: Comparison of computational overhead (numeric results)

Table 5.1: Comparison of computation complexity

Phases	User Registration		Data Reporting		Report Feedback		Reputation Updating	
	User	MSP	User	MSP	User	MSP	User	MSP
Ni et al. [102]	$8\mathcal{E}+2\mathcal{P}$	$8\mathcal{E}$	$(3\kappa+9)\mathcal{E}+(2\kappa+9)\mathcal{P}$	$(4\kappa+10)\mathcal{E}+(3\kappa+9)\mathcal{P}$	$9\mathcal{E}+9\mathcal{P}$	$9\mathcal{E}+10\mathcal{P}$	$16\mathcal{E}+11\mathcal{P}$	$16\mathcal{E}+10\mathcal{P}$
Hartung et al. [103]	$8\mathcal{E}+5\mathcal{P}$	$9\mathcal{E}$	$(3\kappa+9)\mathcal{E}+(2\kappa+7)\mathcal{P}$	$(4\kappa+8)\mathcal{E}+(3\kappa+9)\mathcal{P}$	$9\mathcal{E}+7\mathcal{P}$	$8\mathcal{E}+9\mathcal{P}$	$15\mathcal{E}+12\mathcal{P}$	$16\mathcal{E}+9\mathcal{P}$
Our Simplified Scheme	$8\mathcal{E}+2\mathcal{P}$	$8\mathcal{E}$	$(4\kappa+11)\mathcal{E}+7\mathcal{P}$	$(\kappa+18)\mathcal{E}+7\mathcal{P}$	$8\mathcal{E}+7\mathcal{P}$	$8\mathcal{E}+7\mathcal{P}$	$15\mathcal{E}+9\mathcal{P}$	$15\mathcal{E}+7\mathcal{P}$

Table 5.2: Comparison of communication complexity

Phases	User Registration	Data Reporting	Report Feedback	Reputation Updating
Ni et al. [102]	$4\|Z_p\|+3\|G\|$	$8\|Z_p\|+2\|G\|+(\kappa+1)\|\mathcal{G}\|$	$8\|Z_p\|+2\|G\|+\|\mathcal{G}\|$	$12\|Z_p\|+5\|G\|+\|\mathcal{G}\|$
Hartung et al. [103]	$4\|Z_p\|+4\|G\|$	$5\|Z_p\|+4\|G\|+(\kappa+1)\|\mathcal{G}\|$	$5\|Z_p\|+4\|G\|+\|\mathcal{G}\|$	$9\|Z_p\|+7\|G\|+\|\mathcal{G}\|$
Our Simplified Scheme	$5\|Z_p\|+2\|G\|$	$(2\kappa+10)\|Z_p\|+4\|G\|+\|\mathcal{G}\|$	$5\|Z_p\|+3\|G\|+\|\mathcal{G}\|$	$10\|Z_p\|+5\|G\|+\|\mathcal{G}\|$

scheme's communication overhead is similar to the existing schemes. Although our scheme is not the most efficient one, they are still acceptable.

5.5.2 Implementation & Experiment Results

To demonstrate the feasibility, we develop a proof-of-concept prototype in a Macbook laptop and an Android smartphone. The configuration of the laptop is as mentioned below: 3.1 GHz Dual-Core Intel Core i7 and 16 GB 1867 MHz DDR3, and the configuration of the smartphone is as mentioned blow: Kirin 980 2.6Ghz and 6GH RAM. All algorithms and protocols of our scheme are implemented in the prototype, which is developed based on Java language and a Java-based pairing-based library (JPBC) [147]. Due to Java's cross-platform property, the prototype can be easily ported from a laptop Java application to an android application. According to the JPBC, we choose a type-f curve (a Barreto-Naehrig curve) with the security level $\lambda = 160$ and the embedding degree 12, which supports an asymmetric type-3 bilinear map. We also set $\kappa = 10$, which indicates the range of the reputation score is $[0, 1023]$.

For each phase, we run the prototype 100 times to obtain the average running time at user side and MSP side on the laptop, and the experimental results are shown in Figure 5.4. It only takes less than 1 seconds for a user to achieve user registration phase (Phase-1), report feedback phase (Phase-3), and reputation updating phase (Phase-4), while the data reporting phase (Phase-2) bears large computational costs. The reason is that a user needs to prove two zero-knowledge range proofs in this phase, and the cost of a range proof is decided by the parameter κ, according to our analysis. Since most of the users may utilize the road condition monitoring applications on the smartphone, we also run the prototype 100 times to obtain the average running time at user side on the smartphone, and the experimental results are shown in Table 5.3. However, due to the limited computational capability of the smartphone, the computational delay is larger but the user can pre-compute partial values offline, especially the zero-knowledge proofs, to significantly reduce the computational delay. If the partial zero-knowledge proofs are pre-computed, the computation delay can be less than 1 second. During the user registration phase and the reputation updating phase, if n' is large, i.e., the number of used serial numbers (the size of SList) is large, the delay of computing the witnesses wit_1 and wit_2 is large. Nevertheless, computing these witnesses will not affect the performance of our scheme since the witnesses can be calculated offline. In addition, the communication overhead is shown in Table 5.4. The largest communication data is less than 3 KB, which is acceptable according to the networking performance of current wireless networks.

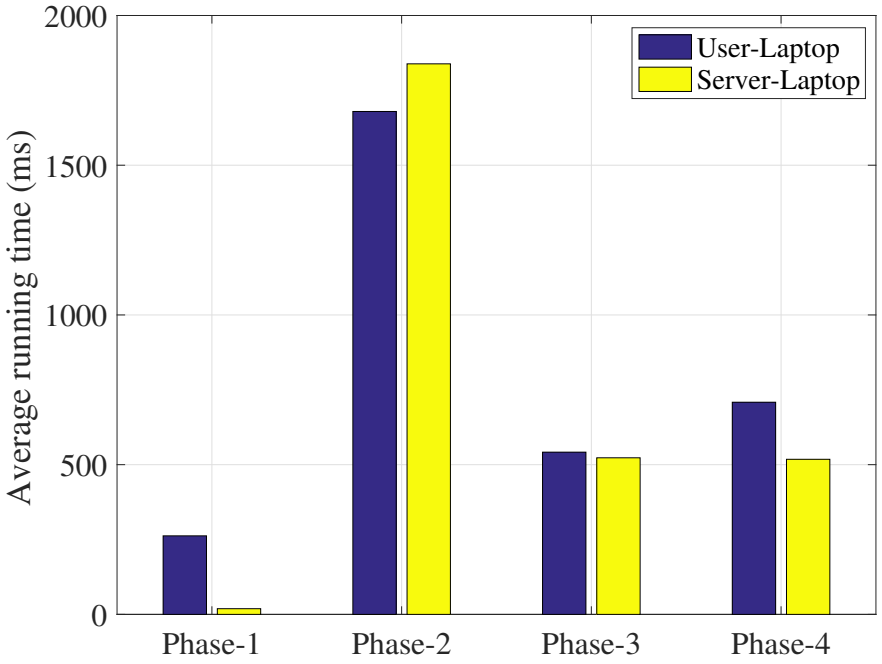


Figure 5.4: Computation costs of the prototype on laptop: User Registration (Phase-1), Data Reporting (Phase-2), Report Feedback (Phase-3), Reputation Updating (Phase-4).

Table 5.3: Computation costs of the prototype on smartphone at user side (unit: second)

Phases	User Registration	Data Reporting
Delay	15.2313	100.7739
Phases	Report Feedback	Reputation Updating
Delay	30.6503	44.3577

Table 5.4: Communication overhead of the prototype (unit: bytes)

Phases	User Registration	Data Reporting
Size	630	2800
Phases	Report Feedback	Reputation Updating
Size	860	1610

5.6 Summary

In this chapter, we focus on the scenario of crowdsourcing-based road condition monitoring and have proposed a novel privacy-preserving monitoring scheme that supports anonymous user reputation management. The proposed scheme provides strong protection of users' privacy via anonymizing users' identities and users can self-maintain their reputation scores without revealing them to a monitoring service provider. Since the proposed scheme does not require any participation of third parties, it is realistic for real-world crowdsourcing applications. We have also given detailed security analysis of our scheme and implemented a prototype to demonstrate the feasibility of our scheme.

Chapter 6

Conclusions and Future Work

In this thesis, we have investigated privacy-preserving mechanisms for V2X services, especially focusing on identity privacy. By resolving the conflict between privacy and availability from three aspects: privacy vs. linkability, privacy vs. accountability, and privacy vs. reliability, a suite of effective privacy-preserving mechanisms have been proposed. In the following, we summarize the main contributions of this thesis, introduce future research directions, and give some final remarks.

6.1 Conclusions

The major contributions of this thesis are summarized as follows.

- First, we have studied the conflict between privacy and linkability for V2X services. To clearly illustrate the conflict issue, we have focused on a specific automated valet parking (AVP) service and analyzed the privacy requirements of users and the potential attacks on the service. Based on the analysis, we have formalized a security model and proposed a novel privacy-preserving parking reservation scheme for an AVP service. In the proposed scheme, we have leveraged BBS+ signature to design an anonymous authentication protocol to protect user privacy during the parking reservation process. Moreover, we have designed several zero-knowledge proof protocols and utilize a proxy re-signature technique to guarantee that a user cannot reserve new parking slot before releasing his/her last parking slot. That is, a parking service provider can resist against "Double-Reservation Attack". In addition, we have

given detailed security analysis to demonstrate that the proposed scheme achieves all desirable security and privacy properties.

- Second, we have studied the conflict between privacy and accountability for V2X services. To clearly illustrate the conflict issue, we have focused on a specific car sharing service. To overcome the issue of single point of failure and avoid the deployment of a centralized trusted authority, we have introduced dynamic and decentralized validation servers to provide decentralized trust for users. Based on the architecture, we have designed a novel privacy-preserving identity management scheme. Specifically, by exploiting a verifiable secret sharing technique and cryptographic accumulators, we have designed several zero-knowledge proof protocols to guarantee that correct users' identities are hidden but can be traced by distributed validation servers. Furthermore, we have designed a dynamic identity transferring protocol to ensure that validation servers can be dynamically changed over time, i.e., a validation server is only responsible for maintaining users' identities for a short period time such that it is more difficult for adversaries to compromise user privacy to achieve higher security guarantees. In addition, we have utilized a simulation-based method to prove that the proposed scheme achieves user privacy preservation and accountability.

- Third, we have studied the conflict between privacy and reliability for V2X services. To clearly illustrate the conflict issue, we have focused on a specific crowdsourcing-based road condition monitoring service. To preserve user privacy, we have designed a privacy-preserving crowdsourcing-based road condition monitoring scheme based on PS signature and zero-knowledge proof. Without relying on a centralized trusted authority to maintain the reputation scores for users, we have additionally designed a novel reputation management and updating mechanism based on homomorphic commitments, cryptographic accumulators and pseudo-random functions, where only users self-maintain their reputation scores. A new security property named K-tolerant trust has been defined in the proposed scheme, to ensure that a user cannot make more that K reports before updating his/her reputation score, i.e., his/her reputation score is the latest. In addition, we have thoroughly analyzed the security properties of the proposed scheme to demonstrate that the proposed scheme achieve user privacy preservation and reliability.

6.2 Future Research Directions

This thesis introduces the V2X communication architecture and its services, identifies privacy challenges in V2X services, and proposes several promising privacy-preserving mechanisms to protect user privacy. As the thesis mainly focuses on the identity privacy of users, there are still open research directions including but not limited to the following three topics.

6.2.1 Blockchain-based Data Management for V2X Services with Privacy Regulation Compliance

With the development of V2X communications, a large amount of V2X services will appear and more data is expected to be generated and maintained by not only users and service providers but also various network operators and government departments. These entities have strong motivations to share their data to boost their service quality and bring more effective services to users. For example, a car insurance service provider can share its user's information with a car sharing service provider such that the user can utilize his/her insurance to cover the insurance of a shared car when enjoying the car sharing service. However, these entities do not trust each other, as there exist many data leakage events caused by data sharing. For instance, one of the biggest network operator in Canada, Rogers, suffers from a serious data leakage caused by external service providers [1]. Under the circumstances, a new data management architecture is required as sensitive data collected from users may leak to more entities and cause more serious privacy concerns. Moreover, privacy laws, e.g., General Data Protection Regulation, have been published to regulate the process of data management by defining data controllers and data processors. After the data is uploaded by users, how these data can be utilized in V2X services should strictly comply with these privacy laws. Facing these two issues, to cross trust boundaries among different entities and to provide data privacy protection in the meantime, blockchain technology can be introduced to manage the decentralized data owned by different entities. Blockchain has many useful characteristics, including immunity, transparency, and decentralization, which are particularly fit for data management and data sharing scenarios for V2X services. Moreover, smart contracts deployed in the blockchain can be automatically run. Hence, they are suitable for writing privacy laws and all data-related operations can automatically verified on the smart contracts to comply with privacy laws. Nevertheless, a blockchain-based data management architecture still

[1] `https://www.rogers.com/support/10022020`

faces many challenges. First, the blockchain has a scalability issue since it needs to run a consensus protocol to reach a system consistence among all participators. Traditional consensus protocols, such as proof-of-work protocol, proof of stake protocol, and byzantine fault tolerance protocol, will lead to a large delay when there exist many participators. Second, orginal public blockchain architectures such as bitcoin and ethereum expose all data stored on chain publicly. Namely, public blockchains do not support privacy preservation and consortium blockchains are more appropriate for achieve data management. Therefore, in the future work, we aim to propose a blockchain-based data management architecture for V2X service, which is fully compatible with privacy regulations.

6.2.2 Verifiable and Privacy-Preserving Federated Learning for V2X Services

With the popularity of V2X services, machine learning and artificial intelligence will be introduced to improve the service quality. Among all machine learning approaches, federated learning is a distributed machine learning approach which enables model training on a large corpus of decentralized data from mobile users. In V2X services, mobile devices owned by users, such as smart vehicles and smartphones, can collaboratively learn a shared prediction model while keeping all the training data on the device without the assistance of powerful cloud servers. Different from traditional machine learning approaches, although the user is just required to run a local machine learning algorithm and exchange some training parameters with other participators, the cooperation among different mobile devices may still raise privacy concerns and users may be not willing to participate in a federated learning for V2X services. Under this circumstance, privacy-preserving mechanisms such as secure multi-party computation and differential privacy can be applied to provide privacy preservation for users. Secure multi-party computation approaches like homomorphic encryption and garble circuits enable users to contribute their data in a ciphertext format and only final results are publicly revealed. These approaches can be integrated with federated learning scenarios in V2X services to ensure the confidentiality of individual's data and parameters. From another point of view, differential privacy protects individual user privacy by adding noises and only the noised data is shared with other users to achieve privacy-preserving federated learning. However, these privacy-preserving mechanisms are not sufficient as they may degrade the performance of the federated learning. Secure multi-party computation requires time-consuming computations and multiple rounds of communication which may cause tremendous time delays. Differential privacy, compared with secure multi-party computation is more lightweight, but it may cause the final result of model training to be inaccurate since a lot of noises have been added to

the model. In addition, as the machine learning model is trained through multiple mobile devices, the final training results should support to be verified to ensure that all contributors have contributed correct parameters and data during the model training process. The verification becomes more difficult when privacy-preserving mechanisms are applied. Therefore, in the future work, we aim to propose a verifiable and privacy-preserving federated learning architecture for V2X services, which can assist different V2X services and their users to corporately achieve the training of machine learning model .

6.2.3 Location Privacy Protection Enhancement in V2X Services

Current works in this thesis mainly focus on protecting the identity privacy of users in V2X Services. Although hiding users' identities can somehow protect users' location privacy by cutting off the linkage between locations, it is not sufficient for V2X services since some V2X services require users to continuously report their locations. As there exist hidden connections between two continuous locations, adversaries have a great chance to link two locations based on a user's moving speed and the restrictions of a road network. Therefore, some location privacy protection mechanisms have been proposed to address this issue, such as location perturbation, spatial cloaking, dummy location. For a location-perturbation-based method, some noises are added to users' original locations, and thus adversaries cannot extract the real locations from the noised locations. The performance of this method is fully controlled by the degree of added noises. On one hand, if the added noises are too huge, the system utility related to location-based V2X services will be decreased. On the other hand, If the noises are too small, the location privacy cannot be guaranteed. For a spatial-cloaking-based method, user can increase the granularity of his/her locations such that the real locations can be hidden inside a cloaked region. By doing so, adversaries can hardly distinguish the target location from a region. Obviously, this location privacy protection mechanism is not suitable for the V2X service that requires accurate location data, such as traffic monitoring services. For a dummy-based method, users can generate dummy locations and mix his/her real locations with dummy locations to protect their location privacy. The dummy size is an important parameter that determines the privacy level of users and the performance of the system. Note that, the above-mentioned location privacy protection mechanisms have a common characteristic, i.e., users need to balance the privacy and system utility by changing some parameters. However, there does not exist a widely accepted approach to help user determine what privacy level they need in different situations. Therefore, in a future work, we aim to propose a scenario-adapted location privacy protection scheme that combines all these three types of privacy-preserving mechanisms. The location privacy level of users should be measured according to the

service type, current locations, historical information, which can help users to make a better decision.

6.3 Final Remarks

In this thesis, we have presented a set of privacy-preserving mechanisms for V2X services, and identified three further research directions to encourage successive research efforts and complement of this thesis. In addition, to further demonstrate the feasibility of our research accomplishments, we will also seek industrial corporations to develop real-world products to further confirm our research findings.

References

[1] J. B. Kenney, "Dedicated short-range communications (DSRC) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[2] H. Seo, K. Lee, S. Yasukawa, Y. Peng, and P. J. Sartori, "LTE evolution for vehicle-to-everything services," *IEEE Communications Magazine*, vol. 54, no. 6, pp. 22–28, 2016.

[3] G. Naik, B. Choudhury, and J. Park, "IEEE 802.11bd & 5g NR V2X: evolution of radio access technologies for V2X communications," *IEEE Access*, vol. 7, pp. 70 169–70 184, 2019.

[4] S. A. Ashraf, R. Blasco, H. Do, G. Fodor, C. Zhang, and W. Sun, "Supporting vehicle-to-everything services by 5g new radio release-16 systems," *IEEE Communications Standards Magazine*, vol. 4, no. 1, pp. 26–32, 2020.

[5] S. Karnouskos and F. Kerschbaum, "Privacy and integrity considerations in hyper-connected autonomous vehicles," *Proceedings of the IEEE*, vol. 106, no. 1, pp. 160–170, 2018.

[6] R. Lu, L. Zhang, J. Ni, and Y. Fang, "5g vehicle-to-everything services: Gearing up for security and privacy," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 373–389, 2020.

[7] J. Huang, D. Fang, Y. Qian, and R. Q. Hu, "Recent advances and challenges in security and privacy for v2x communications," *IEEE Open Journal of Vehicular Technology*, vol. 1, pp. 244–266, 2020.

[8] 3GPP, "Nr; overall description;," *3GPP TS 38.300*, 2020.

[9] X. Wang, S. Mao, and M. X. Gong, "An overview of 3gpp cellular vehicle-to-everything standards," *GetMobile: Mobile Computing and Communications*, vol. 21, no. 3, pp. 19–25, 2017.

[10] R. Dang, J. Ding, B. Su, Q. Yao, Y. Tian, and K. Li, "A lane change warning system based on V2V communication," in *17th International IEEE Conference on Intelligent Transportation Systems*, 2014, pp. 1923–1928.

[11] N. Mirnig, N. Perterer, G. Stollnberger, and M. Tscheligi, "Three strategies for autonomous car-to-pedestrian communication: A survival guide," in *Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 2017, pp. 209–210.

[12] C. Huang, R. Lu, and K. R. Choo, "Vehicular fog computing: Architecture, use case, and security and forensic challenges," *IEEE Communications Magazine*, vol. 55, no. 11, pp. 105–111, 2017.

[13] A. Pham, I. Dacosta, B. Jacot-Guillarmod, K. Huguenin, T. Hajar, F. Tramèr, V. Gligor, and J.-P. Hubaux, "Privateride: A privacy-enhanced ride-hailing service," *Proceedings of Privacy Enhancing Technologies*, vol. 2017, no. 2, pp. 38–56, 2017.

[14] R. Neuhaus, E. Lenz, S. S. Borojeni, and M. Hassenzahl, "Exploring the future experience of automated" valet parking"-a user enactment," in *Proceedings of the 11th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, 2019, pp. 24–34.

[15] M. Cocca, D. Giordano, M. Mellia, and L. Vassio, "Free floating electric car sharing: A data driven approach for system design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 12, pp. 4691–4703, 2019.

[16] X. Wang, X. Zheng, Q. Zhang, T. Wang, and D. Shen, "Crowdsourcing in its: The state of the work and the networking," *IEEE Transactions on Intelligent Transportation systems*, vol. 17, no. 6, pp. 1596–1605, 2016.

[17] IEEE, "Ieee standard for wireless access in vehicular environments–security services for applications and management messages," *IEEE 1609.2*, pp. 1–240, 2017.

[18] 3GPP, "Security aspect for lte support of vehicle-to-everything (v2x) services," *3GPP TS 33.185*, pp. 1–10, 2017.

[19] S. Chhabra, "University of alberta receive 500,000 dollars in connected vehicle research funding," https://mobilesyrup.com/2018/06/25/university-of-alberta-receives-500000-in-connected-vehicle-research-funding, 2018, [Online; accessed 27-June-2018].

[20] S. Melendez and A. Pasternack, "Here are the data brokers quietly buying and selling your personal information," https://www.fastcompany.com/90310803/here-are-the-data-brokers-quietly-buying-and-selling-your-personal-information, 2019, [Online; accessed 27-April-2020].

[21] J. Katz and Y. Lindell, *Introduction to modern cryptography*. Boca Raton, FL: CRC press, 2014.

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

[23] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Proceedings of 24th Annual International Cryptology Conference*, 2004, pp. 41–55.

[24] J. Camenisch, S. Hohenberger, and A. Lysyanskaya, "Compact e-cash," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, vol. 3494. Springer, 2005, pp. 302–321.

[25] B. Libert and D. Vergnaud, "Multi-use unidirectional proxy re-signatures," in *Proceedings of the 2008 ACM Conference on Computer and Communications Security*, 2008, pp. 511–520.

[26] D. Pointcheval and O. Sanders, "Short randomizable signatures," in *Cryptographers' Track at the RSA Conference*, ser. Lecture Notes in Computer Science, vol. 9610, 2016, pp. 111–126.

[27] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Applied Mathematics*, vol. 156, no. 16, pp. 3113–3121, 2008.

[28] Y. Dodis and A. Yampolskiy, "A verifiable random function with short proofs and keys," in *International Workshop on Public Key Cryptography*, 2005, pp. 416–431.

[29] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Proceedings of Annual International Cryptology Conference*, 1991, pp. 129–140.

[30] M. H. Au, W. Susilo, Y. Mu, and S. S. M. Chow, "Constant-size dynamic k -times anonymous authentication," *IEEE Systems Journal*, vol. 7, no. 2, pp. 249–261, 2013.

[31] D. Derler, C. Hanser, and D. Slamanig, "Revisiting cryptographic accumulators, additional properties and relations to other primitives," in *Cryptographers' track at the rsa conference*, vol. 9048, 2015, pp. 127–144.

[32] J. Li, N. Li, and R. Xue, "Universal accumulators with efficient nonmembership proofs," in *International Conference on Applied Cryptography and Network Security*, vol. 4521, 2007, pp. 253–269.

[33] 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 *Cryptographers' track at the RSA conference*, vol. 5473.

[34] O. Goldreich and Y. Oren, "Definitions and properties of zero-knowledge proof systems," *Journal of Cryptology*, vol. 7, no. 1, pp. 1–32, 1994.

[35] D. Bernhard, O. Pereira, and B. Warinschi, "How not to prove yourself: Pitfalls of the fiat-shamir heuristic and applications to helios," in *International Conference on the Theory and Application of Cryptology and Information Security.*, vol. 7658, 2012, pp. 626–643.

[36] I. Wagner and D. Eckhoff, "Technical privacy metrics: A systematic survey," *ACM Computing Surveys*, vol. 51, no. 3, pp. 57:1–57:38, 2018.

[37] A. Pfitzmann and M. Köhntopp, "Anonymity, unobservability, and pseudonymity - A proposal for terminology," in *Proceedings of Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability*, 2000, pp. 1–9.

[38] C. Díaz, "Anonymity metrics revisited," in *Anonymous Communication and its Applications, 09.10. - 14.10.2005*, 2005.

[39] J. Hubaux, S. Capkun, and J. Luo, "The security and privacy of smart vehicles," *IEEE Security & Privacy*, vol. 2, no. 3, pp. 49–55, 2004.

[40] B. Wiedersheim, Z. Ma, F. Kargl, and P. Papadimitratos, "Privacy in inter-vehicular networks: Why simple pseudonym change is not enough," in *Wireless On-demand Network Systems and Services (WONS), Seventh International Conference on.* IEEE, 2010, pp. 176–183.

[41] D. Eckhoff, C. Sommer, T. Gansen, R. German, and F. Dressler, "Strong and affordable location privacy in vanets: Identity diffusion using time-slots and swapping," in *Proceedings of the Second IEEE Vehicular Networking Conference*, 2010, pp. 174–181.

[42] D. Eckhoff, R. German, C. Sommer, F. Dressler, and T. Gansen, "Slotswap: strong and affordable location privacy in intelligent transportation systems," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 126–133, 2011.

[43] Y. Pan, J. Li, L. Feng, and B. Xu, "An analytical model for random pseudonym change scheme in vanets," *Cluster Computing*, vol. 17, no. 2, pp. 413–421, 2014.

[44] K. Sampigethaya, L. Huang, M. Li, R. Poovendran, K. Matsuura, and K. Sezaki, "Caravan: Providing location privacy for vanet," Washington Univ Seattle Dept Of Electrical Engineering, Tech. Rep., 2005.

[45] K. Sampigethaya, M. Li, L. Huang, and R. Poovendran, "AMOEBA: robust location privacy scheme for VANET," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 8, pp. 1569–1589, 2007.

[46] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. Shen, "Anonymity analysis on social spot based pseudonym changing for location privacy in vanets," in *Proceedings of IEEE International Conference on Communications, ICC 2011, Kyoto, Japan, 5-9 June, 2011*, 2011, pp. 1–5.

[47] R. Lu, X. Lin, T. H. Luan, X. Liang, and X. S. Shen, "Pseudonym changing at social spots: An effective strategy for location privacy in vanets," *IEEE Transactions Vehicular Technology*, vol. 61, no. 1, pp. 86–96, 2012.

[48] R. Yu, J. Kang, X. Huang, S. Xie, Y. Zhang, and S. Gjessing, "Mixgroup: Accumulative pseudonym exchanging for location privacy enhancement in vehicular social networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 1, pp. 93–105, 2016.

[49] K. Emara, W. Woerndl, and J. H. Schlichter, "Context-based pseudonym changing scheme for vehicular adhoc networks," *arXiv preprint*, vol. arXiv:1607.07656, 2016.

[50] B. Ying and D. Makrakis, "Reputation-based pseudonym change for location privacy in vehicular networks," in *IEEE International Conference on Communications*, 2015, pp. 7041–7046.

[51] G. Calandriello, P. Papadimitratos, J. Hubaux, and A. Lioy, "Efficient and robust pseudonymous authentication in VANET," in *Proceedings of the Fourth International Workshop on Vehicular Ad Hoc Networks*, 2007, pp. 19–28.

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

[53] X. Lin, X. Sun, X. Wang, C. Zhang, P. Ho, and X. Shen, "TSVC: timed efficient and secure vehicular communications with privacy preserving," *IEEE Transactions on Wireless Communications*, vol. 7, no. 12-1, pp. 4987–4998, 2008.

[54] C. Zhang, X. Lin, R. Lu, P. Ho, and X. Shen, "An efficient message authentication scheme for vehicular communications," *IEEE Transactions on Vehicular Technology.*, vol. 57, no. 6, pp. 3357–3368, 2008.

[55] D. Chaum and E. van Heyst, "Group signatures," in *Proceedings of Workshop on the Theory and Application of of Cryptographic Techniques*, 1991, pp. 257–265.

[56] J. K. Liu, V. K. Wei, and D. S. Wong, "Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract)," in *Proceedings of Information Security and Privacy: 9th Australasian Conference*, 2004, pp. 325–335.

[57] J. Y. Hwang, L. Chen, H. S. Cho, and D. Nyang, "Short dynamic group signature scheme supporting controllable linkability," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 6, pp. 1109–1124, 2015.

[58] R. Lu, X. Lin, H. Zhu, P. Ho, and X. Shen, "ECPP: efficient conditional privacy preservation protocol for secure vehicular communications," in *27th IEEE International Conference on Computer Communications*, 2008, pp. 1229–1237.

[59] T. Nakanishi and N. Funabiki, "Verifier-local revocation group signature schemes with backward unlinkability from bilinear maps," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. 90-A, no. 1, pp. 65–74, 2007.

[60] V. Kumar, H. Li, N. Luther, P. Asokan, J. J. Park, K. Bian, M. B. H. Weiss, and T. Znati, "Direct anonymous attestation with efficient verifier-local revocation for subscription system," in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018, pp. 567–574.

[61] F. Baldimtsi, J. Camenisch, M. Dubovitskaya, A. Lysyanskaya, L. Reyzin, K. Samelin, and S. Yakoubov, "Accumulators with applications to anonymity-preserving revocation," in *2017 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2017, pp. 301–315.

[62] T. Acar, S. S. M. Chow, and L. Nguyen, "Accumulators and u-prove revocation," in *International Conference on Financial Cryptography and Data Security*, A. Sadeghi, Ed., vol. 7859, 2013, pp. 189–196.

[63] G. Yan, W. Yang, D. B. Rawat, and S. Olariu, "Smartparking: A secure and intelligent parking system," *IEEE intelligent transportation systems magazine*, vol. 3, no. 1, pp. 18–30, 2011.

[64] S. Biswas and J. V. Misic, "Prioritized wave-based parking assistance with security and user anonymity." *Journal of Communications*, vol. 7, no. 8, pp. 577–586, 2012.

[65] R. Lu, X. Lin, H. Zhu, and X. Shen, "Spark: A new vanet-based smart parking scheme for large parking lots," in *Proceedings of IEEE INFOCOM*. IEEE, 2009, pp. 1413–1421.

[66] ——, "An intelligent secure and privacy-preserving parking scheme through vehicular communications," *IEEE Transactions on Vehicular Technology*, vol. 59, no. 6, pp. 2772–2785, 2010.

[67] J. Ni, K. Zhang, X. Lin, Y. Yu, and X. S. Shen, "Cloud-based privacy-preserving parking navigation through vehicular communications," in *Proceedings of Security and Privacy in Communication Networks - 12th International Conference*, 2016, pp. 85–103.

[68] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. S. Shen, "Privacy-preserving smart parking navigation supporting efficient driving guidance retrieval," *IEEE Transactions Vehicular Technology*, to appear.

[69] R. Garra, S. Martínez, and F. Sebé, "A privacy-preserving pay-by-phone parking system," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 5697–5706, 2017.

[70] R. Borges and F. Sebé, "Parking tickets for privacy-preserving pay-by-phone parking," in *Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society.*, 2019, pp. 130–134.

[71] J. Ni, X. Lin, and X. Shen, "Toward privacy-preserving valet parking in autonomous driving era," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2893–2905, 2019.

[72] I. Chatzigiannakis, A. Vitaletti, and A. Pyrgelis, "A privacy-preserving smart parking system using an iot elliptic curve based security platform," *Computer Communications*, vol. 89, pp. 165–177, 2016.

[73] J. Hu, D. He, Q. Zhao, and K.-K. R. Choo, "Parking management: A blockchain-based privacy-preserving system," *IEEE Consumer Electronics Magazine*, vol. 8, no. 4, pp. 45–49, 2019.

[74] L. Wang, X. Lin, E. Zima, and C. Ma, "Towards airbnb-like privacy-enhanced private parking spot sharing based on blockchain," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 2411–2423, 2020.

[75] W. Al Amiri, M. Baza, K. Banawan, M. Mahmoud, W. Alasmary, and K. Akkaya, "Privacy-preserving smart parking system using blockchain and private information retrieval," in *2019 International Conference on Smart Applications, Communications and Networking (SmartNets)*, 2019, pp. 1–6.

[76] S. Ahmed, M. S. Rahman, M. S. Rahaman *et al.*, "A blockchain-based architecture for integrated smart parking systems," in *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2019, pp. 177–182.

[77] Y. Kanza and E. Safra, "Cryptotransport: blockchain-powered ride hailing while preserving privacy, pseudonymity and trust," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2018, pp. 540–543.

[78] J. Ni, K. Zhang, X. Lin, H. Yang, and X. S. Shen, "Ama: Anonymous mutual authentication with traceability in carpooling systems," in *2016 IEEE International Conference on Communications (ICC)*, 2016, pp. 1–6.

[79] P. Hallgren, C. Orlandi, and A. Sabelfeld, "Privatepool: privacy-preserving ridesharing," in *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, 2017, pp. 276–291.

[80] Q. Zhao, C. Zuo, G. Pellegrino, and L. Zhiqiang, "Geo-locating drivers: A study of sensitive data leakage in ride-hailing services," in *Annual Network and Distributed System Security symposium*, 2019.

[81] Y. Khazbak, J. Fan, S. Zhu, and G. Cao, "Preserving location privacy in ride-hailing service," in *2018 IEEE Conference on Communications and Network Security (CNS)*, 2018, pp. 1–9.

[82] P. Goel, L. Kulik, and K. Ramamohanarao, "Optimal pick up point selection for effective ride sharing," *IEEE Transactions on Big Data*, vol. 3, no. 2, pp. 154–168, 2016.

[83] ——, "Privacy-aware dynamic ride sharing," *ACM Transactions on Spatial Algorithms and Systems*, vol. 2, no. 1, pp. 1–41, 2016.

[84] Y. Luo, X. Jia, S. Fu, and M. Xu, "pride: Privacy-preserving ride matching over road networks for online ride-hailing service," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1791–1802, 2018.

[85] I. Symeonidis, M. A. Mustafa, and B. Preneel, "Keyless car sharing system: A security and privacy analysis." in *IEEE International Smart Cities Conference*, 2016, pp. 1–7.

[86] 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 *Proceedings of European Symposium on Research in Computer Security*, 2017, pp. 475–493.

[87] P. Ananth, A. R. Choudhuri, A. Goel, and A. Jain, "Round-optimal secure multiparty computation with honest majority," in *Proceedings of Annual International Cryptology Conference*, 2018, pp. 395–424.

[88] A. Dmitrienko and C. Plappert, "Secure free-floating car sharing for offline cars," in *Proceedings of ACM on Conference on Data and Application Security and Privacy*, 2017, pp. 349–360.

[89] M. Li, L. Zhu, and X. Lin, "Privacy-preserving traffic monitoring with false report filtering via fog-assisted vehicular crowdsensing," *IEEE Transactions on Services Computing*, to appear.

[90] S. Basudan, X. Lin, and K. Sankaranarayanan, "A privacy-preserving vehicular crowdsensing-based road surface condition monitoring system using fog computing," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 772–782, 2017.

[91] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, and H. Wang, "Privacy-preserving cloud-based road condition monitoring with source authentication in vanets," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 7, pp. 1779–1790, 2019.

[92] P. Zhou, W. Chen, S. Ji, H. Jiang, L. Yu, and D. O. Wu, "Privacy-preserving online task allocation in edge-computing-enabled massive crowdsensing," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 7773–7787, 2019.

[93] C. Zhang, L. Zhu, J. Ni, C. Huang, and X. Shen, "Verifiable and privacy-preserving traffic flow statistics for advanced traffic management systems," *IEEE Transactions on Vehicular Technology*, to appear.

[94] Q. Kong, L. Su, and M. Ma, "Achieving privacy-preserving and verifiable data sharing in vehicular fog with blockchain," *IEEE Transactions on Intelligent Transportation Systems*, to appear.

[95] J. Zhang, Q. Zhang, and S. Ji, "A fog-assisted privacy-preserving task allocation in crowdsourcing," *IEEE Internet of Things Journal*, to appear.

[96] L. Zhu, C. Zhang, C. Xu, and K. Sharif, "Rtsense: Providing reliable trust-based crowdsensing services in CVCC," *IEEE Network*, vol. 32, no. 3, pp. 20–26, 2018.

[97] M. Li, J. Weng, A. Yang, W. Lu, Y. Zhang, L. Hou, J. Liu, Y. Xiang, and R. H. Deng, "Crowdbc: A blockchain-based decentralized framework for crowdsourcing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 30, no. 6, pp. 1251–1266, 2019.

[98] Y. Yu, S. Liu, L. Guo, P. L. Yeoh, B. Vucetic, and Y. Li, "Crowdr-fbc: A distributed fog-blockchains for mobile crowdsourcing reputation management," *IEEE Internet of Things Journal*, to appear.

[99] H. Wu, L. Wang, G. Xue, J. Tang, and D. Yang, "Enabling data trustworthiness and user privacy in mobile crowdsensing," *IEEE/ACM Transactions on Networking*, vol. 27, no. 6, pp. 2294–2307, 2019.

[100] E. Zhai, D. I. Wolinsky, R. Chen, E. Syta, C. Teng, and B. Ford, "Anonrep: Towards tracking-resistant anonymous reputation," in *13th USENIX Symposium on Networked Systems Design and Implementation*, 2016, pp. 583–596.

[101] X. Yi, K. Lam, E. Bertino, and F. Rao, "Location privacy-preserving mobile crowd sensing with anonymous reputation," in *European Symposium on Research in Computer Security*, vol. 11736, 2019, pp. 387–411.

[102] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1317–1331, 2020.

[103] G. Hartung, M. Hoffmann, M. Nagel, and A. Rupp, "BBA+: improving the security and applicability of privacy-preserving point collection," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 1925–1942.

[104] J. Blömer, F. Eidens, and J. Juhnke, "Practical, anonymous, and publicly linkable universally-composable reputation systems," in *Cryptographers' Track at the RSA Conference*, vol. 10808, 2018, pp. 470–490.

[105] D. Liu, A. Alahmadi, J. Ni, X. Lin, and X. Shen, "Anonymous reputation system for iiot-enabled retail marketing atop pos blockchain," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3527–3537, 2019.

[106] U. Schwesinger, M. Bürki, J. Timpner, S. Rottmann, L. C. Wolf, L. M. Paz *et al.*, "Automated valet parking and charging for e-mobility," in *IEEE Intelligent Vehicles Symposium*, 2016, pp. 157–164.

[107] D.-B. Company, "When the app parks the vehicle," https://www.daimler.com/innovation/next/when-the-app-parks-the-vehicle.html, 2017, [Online; accessed 11-November-2017].

[108] Z. Technology, "Zongmu showcases its first low-speed high-automation product to ces 2018," http://www.zongmutech.com/en/news/20180109378, 2018, [Online; accessed 25-January-2018].

[109] H. Banzhaf, D. Nienhüser, S. Knoop, and J. M. Zöllner, "The future of parking: A survey on automated valet parking with an outlook on high density parking," in *IEEE Intelligent Vehicles Symposium*, 2017, pp. 1827–1834.

[110] T. M. T. Do and D. Gatica-Perez, "The places of our lives: Visiting patterns and automatic labeling from longitudinal smartphone data," *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 638–648, 2014.

[111] Y. He, J. Ni, X. Wang, B. Niu, F. Li, and X. S. Shen, "Privacy-preserving partner selection for ride-sharing services," *IEEE Transactions Vehicular Technology*, to appear.

[112] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Scientific reports*, vol. 3, 2013.

[113] K. Y. Yu, T. H. Yuen, S. S. M. Chow, S. Yiu, and L. C. K. Hui, "PE(AR)2: privacy-enhanced anonymous authentication with reputation and revocation," in *Proceedings of 17th European Symposium on Research in Computer Security*, 2012, pp. 679–696.

[114] M. H. Au, A. Kapadia, and W. Susilo, "BLACR: ttp-free blacklistable anonymous credentials with reputation," in *19th Annual Network and Distributed System Security Symposium*, 2012.

[115] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Linkable ring signature with unconditional anonymity," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 1, pp. 157–165, 2014.

[116] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, "Vehicle-to-everything (v2x) services supported by lte-based systems and 5g," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017.

[117] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: differential privacy for location-based systems," in *2013 ACM SIGSAC Conference on Computer and Communications Security*, 2013, pp. 901–914.

[118] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Nature Scientific Reports*, vol. 3, 2013.

[119] Tor Project, "Orbot: Proxy with tor," https://guardianproject.info/apps/orbot/, 2017, [Online; accessed 15-August-2017].

[120] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Conference on the theory and application of cryptographic techniques*, 1986, pp. 186–194.

[121] C. Huang, R. Lu, H. Zhu, J. Shao, A. Alamer, and X. Lin, "EPPD: efficient and privacy-preserving proximity testing with differential privacy techniques," in *IEEE International Conference on Communications*, 2016, pp. 1–6.

[122] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Proceedings of 24th Annual International Cryptology Conference*, 2004, pp. 56–72.

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

[124] S. Weikl and K. Bogenberger, "Relocation strategies and algorithms for free-floating car sharing systems," *IEEE Intelligent Transportation System Magazine*, vol. 5, no. 4, pp. 100–111, 2013.

[125] 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., 2018.

[126] F. Dandl and K. Bogenberger, "Comparing future autonomous electric taxis with an existing free-floating carsharing system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2037 – 2047, 2019.

[127] L. Hall, "The environmental benefits of car sharing," https://www.fleetcarma.com/environmental-benefits-car-sharing/, 2018, [Online; accessed 15-Janurary-2019].

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

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

[130] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, "Blockchain: A distributed solution to automotive security and privacy," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 119–125, 2017.

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

[132] C. Huang, R. Lu, X. Lin, and X. Shen, "Secure automated valet parking: A privacy-preserving reservation scheme for autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 11 169–11 180, 2018.

[133] L. Zhu, M. Li, Z. Zhang, and Z. Qin, "Asap: An anonymous smart-parking and payment scheme in vehicular networks," *IEEE Transactions on Dependable and Secure Computing*. [Online]. Available: http://dx.doi.org/10.1109/TDSC.2018.2850780

[134] J. Ni, X. Lin, and X. Shen, "Towards privacy-preserving valet parking in autonomous driving era," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2893 – 2905, 2019.

[135] M. Hadian, T. Altuwaiyan, X. Liang, and H. Zhu, "Privacy-preserving task scheduling for time-sharing services of autonomous vehicles," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5260–5270, 2019.

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

[137] A. Madhusudan, I. Symeonidis, M. A. Mustafa, R. Zhang, and B. Preneel, "Sc2share: Smart contract for secure car sharing," in *Proceedings of International Conference on Information Systems Security and Privacy*, 2019, pp. 163–171.

[138] A. C. Hernandez, J. Castellà-Roca, and A. Viejo, "Key management system for private car-sharing scenarios," in *Proceedings of IEEE Vehicular Technology Conference*, 2018, pp. 1–7.

[139] I. Damgård, V. Pastro, N. P. Smart, and S. Zakarias, "Multiparty computation from somewhat homomorphic encryption," in *Proceedings of Annual International Cryptology Conference*, 2012, pp. 643–662.

[140] 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 *Proceedings of ACM Conference on Computer and Communications Security*, 2017, pp. 719–728.

[141] E. Ben-Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in *Proceedings of IEEE Symposium on Security and Privacy*, 2014, pp. 459–474.

[142] R. Dingledine, N. Mathewson, and P. F. Syverson, "Tor: The second-generation onion router," in *Proceedings of USENIX Security Symposium*, 2004, pp. 303–320.

[143] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic," in *Proceedings of Annual International Cryptology Conference*, 1999, pp. 148–164.

[144] T. M. Wong, C. Wang, and J. M. Wing, "Verifiable secret redistribution for archive system," in *Proceedings of IEEE Security in Storage Workshop*, 2002, pp. 94–106.

[145] Z. Wang, X. Luo, and Q. Wu, "Verifiably encrypted group signatures," in *Proceedings of International Conference on Provable Security*, 2017, pp. 107–126.

[146] W. Lueks, M. H. Everts, and J. Hoepman, "Vote to link: Recovering from misbehaving anonymous users," in *Proceedings of ACM on Workshop on Privacy in the Electronic Society*, 2016, pp. 111–122.

[147] A. De Caro and V. Iovino, "jpbc: Java pairing based cryptography," in *Proceedings of IEEE Symposium on Computers and Communications*, 2011, pp. 850–855.

[148] X. Kong, X. Liu, B. Jedari, M. Li, L. Wan, and F. Xia, "Mobile crowdsourcing in smart cities: Technologies, applications, and future challenges," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8095–8113, 2019.

[149] B. Jan, H. Farman, M. Khan, M. Talha, and I. U. Din, "Designing a smart transportation system: An internet of things and big data approach," *IEEE Wireless Communications*, vol. 26, no. 4, pp. 73–79, 2019.

[150] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Y. Zhao, "Ghost riders: Sybil attacks on crowdsourced mobile mapping services," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1123–1136, 2018.

[151] K. C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang, "All your GPS are belong to us: Towards stealthy manipulation of road navigation systems," in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1527–1544.

[152] Q. Yang and H. Wang, "Toward trustworthy vehicular social networks," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 42–47, 2015.

[153] Z. Tian, X. Gao, S. Su, J. Qiu, X. Du, and M. Guizani, "Evaluating reputation management schemes of internet of vehicles based on evolutionary game theory," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5971–5980, 2019.

[154] S. Agrawal, C. Ganesh, and P. Mohassel, "Non-interactive zero-knowledge proofs for composite statements," in *Proceedings of Annual International Cryptology Conference*, 2018, pp. 643–673.

[155] J. Camenisch, R. Chaabouni, and A. Shelat, "Efficient protocols for set membership and range proofs," in *International Conference on the Theory and Application of Cryptology and Information Security*, vol. 5350, 2008, pp. 234–252.

[156] B. Bünz, J. Bootle, D. Boneh, A. Poelstra, P. Wuille, and G. Maxwell, "Bulletproofs: Short proofs for confidential transactions and more," in *2018 IEEE Symposium on Security and Privacy (SP)*, 2018, pp. 315–334.

[157] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Proceedings of Annual international cryptology conference*, vol. 576, 1991, pp. 129–140.

[158] M. Abe, J. Groth, K. Haralambiev, and M. Ohkubo, "Optimal structure-preserving signatures in asymmetric bilinear groups," in *Proceedings of Annual Cryptology Conference*, vol. 6841. Springer, 2011, pp. 649–666.

List of Publications

Journal Papers

(J1) **C. Huang**, D. Liu, J. Ni, R. Lu, and X. Shen, "Achieving Accountable and Efficient Data Sharing in Industrial Internet of Things", IEEE Trans. on Industrial Informatics, DOI: 10.1109/TII.2020.2982942.

(J2) **C. Huang**, R. Lu, J. Ni, and X. Shen, "DAPA: A Decentralized, Accountable, and Privacy-preserving Architecture for Car Sharing Services", IEEE Trans. on Vehicular Technology, Vol. 69, No. 5, pp. 4869-4882, 2020.

(J3) **C. Huang**, R. Lu, J. Ni, L. Xue and X. Shen, "Prove Your Expense Records: Building Trustworthiness for Anonymous Reviewers on Online Review Platforms", IEEE Trans. on Service Computing, Under Revision.

(J4) **C. Huang**, R. Lu, D. Liu, A. Yang, and X. Shen, "Privacy-preserving Crowdsourcing-based Road Condition Monitoring With Anonymous Reputation Management", ready to submit, 2020.

(J5) L. Xue, D. Liu, **C. Huang**, X. Lin, and X. Shen, "Secure and Privacy-Preserving Decision Tree Classification with Lower Complexity", J. Communications and Information Networks, DOI: 10.23919/JCIN.2020.9055107.

(J6) J. Hao, **C. Huang**, J. Ni, H. Rong, M. Xian, and X. Shen, "Fine-Grained Data Access Control with Attribute-Hiding Policy for Cloud-Based IoT", Computer Network (Elsevier), Vol. 153, pp. 1-10, 2019.

(J7) **C. Huang**, R. Lu, X. Lin, and X. Shen, "Secure Automated Valet Parking: A Privacy-Preserving Reservation Scheme for Autonomous Vehicles", IEEE Trans. on Vehicular Technology, Vol. 67, No. 11, pp. 11169-11180, 2018.

(J8) A. Yang, J. Weng, K. Yang, **C. Huang**, and X. Shen, "Delegating Authentication to Edge: A Decentralized Authentication Architecture for Vehicular Network", IEEE Trans. on Intelligent Transportation Systems, to appear.

(J9) C. Zhang, L. Zhu, J. Ni, **C. Huang**, and X. Shen, "Verifiable and Privacy-Preserving Traffic Flow Statistics for Advanced Traffic Management Systems", IEEE Trans. on Vehicular Technology, to appear.

Conference Papers

(C1) **C. Huang**, J. Ni, R. Lu, and X. Shen, "Exploring Anonymous User Review: Linkability Analysis with Machine Learning", Proc. IEEE Globecom'19, Waikoloa, HI, USA, Dec. 9-13, 2019.

(C2) L. Xue, J. Ni, **C. Huang**, X. Lin, and X. Shen, "Forward Secure and Fine-grained Data Sharing for Mobile Crowdsensing", Proc. 17th International Conference on Privacy, Security and Trust (PST)'19, Fredericton, NB, Canada, August 26-28, 2019.

(C3) J. Hao, **C. Huang**, G. Chen, X. Ming, and X. Shen, "Privacy-Preserving Interest-Ability Based Task Allocation in Crowdsourcing", Proc. IEEE ICC'19, Shanghai, China, May 20-24, 2019.

(C4) **C. Huang**, J. Ni, R. Lu, and X. Shen, "Online Advertising with Verifiable Fairnes", Proc. IEEE ICC'19, Shanghai, China, May 20-24, 2019.

(C5) J. Hao, **C. Huang**, J. Liu, M. Xian, and X. Shen, "Efficient Outsourced Data Access Control with User Revocation for Cloud-based IoT", Proc. IEEE Globecom'18, Abu Dhabi, UAE, Dec. 9-13, 2018.

(C6) **C. Huang**, D. Liu, J. Ni, R. Lu, and X. Shen, "Reliable and Privacy-Preserving Selective Data Aggregation for Fog-Based IoT", Proc. IEEE ICC'18, Kansas City, MO, USA, May 20-24, 2018. (**Best Paper Award**)

Vita

Cheng Huang received the B.Eng degree in Information Security and M.Eng degree in Information Security from the Xidian University, Xi'an, China, in 2013 and 2016, respectively. He was a Project Officer with the INFINITUS laboratory at the School of Electrical and Electronic Engineering, Nanyang Technological University till July 2016. Starting from August 2016, he is working toward the Ph.D. degree with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His current research interests include applied cryptography, privacy preservation for vehicle-based services, and blockchain security. Mr.Huang served as a Technical Program Committee Member for many conferences, including IEEE Globecom'18, IEEE VTC-FALL'19, IEEE ICC'20, and IEEE ICNC'20. He has won best paper awards at the conferences, including IEEE ICCC'16 and IEEE ICC'18, and many prestigious awards, such as Faculty of Engineering Graduate Scholarship Awards and Jon W. Mark Graduate Scholarship.