Security and Privacy Preservation in Mobile Crowdsensing

by

Jianbing Ni

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, 2018

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	Vojislav B. Mišić
	Title	Professor
Supervisor	NAME	Xuemin (Sherman) Shen
	Title	University Professor
Co–Supervisor	NAME	Xiaodong Lin
	Title	Professor
Internal Member	NAME	Liang-Liang Xie
	Title	Professor
Internal Member	NAME	Sagar Naik
	Title	Professor
Internal–external Member	NAME	Jun Liu
	Title	Associate Professor

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

Mobile CrowdSensing (MCS) is a compelling paradigm that enables a crowd of individuals to cooperatively collect and share data to measure phenomena or record events of common interest using their mobile devices. Pairing with inherent mobility and intelligence, mobile users can collect, produce and upload large amounts of data to service providers based on crowdsensing tasks released by customers, ranging from general information, such as temperature, air quality and traffic condition, to more specialized data, such as recommended places, health condition and voting intentions. Compared with traditional sensor networks, MCS can support large-scale sensing applications, improve sensing data trustworthiness and reduce the cost on deploying expensive hardware or software to acquire high-quality data.

Despite the appealing benefits, however, MCS is also confronted with a variety of security and privacy threats, which would impede its rapid development. Due to their own incentives and vulnerabilities of service providers, data security and user privacy are being put at risk. The corruption of sensing reports may directly affect crowdsensing results, and thereby mislead customers to make irrational decisions. Moreover, the content of crowdsensing tasks may expose the intention of customers, and the sensing reports might inadvertently reveal sensitive information about mobile users. Data encryption and anonymization techniques can provide straightforward solutions for data security and user privacy, but there are several issues, which are of significantly importance to make MCS practical. First of all, to enhance data trustworthiness, service providers need to recruit mobile users based on their personal information, such as preferences, mobility pattern and reputation, resulting in the privacy exposure to service providers. Secondly, it is inevitable to have replicate data in crowdsensing reports, which may possess large communication bandwidth, but traditional data encryption makes replicate data detection and deletion challenging. Thirdly, crowdsensed data analysis is essential to generate crowdsensing reports in MCS, but the correctness of crowdsensing results in the absence of malicious mobile users and service providers become a huge concern for customers. Finally yet importantly, even if user privacy is preserved during task allocation and data collection, it may still be exposed during reward distribution. It further discourage mobile users from task participation.

In this thesis, we explore the approaches to resolve these challenges in MCS. Based on the architecture of MCS, we conduct our research with the focus on security and privacy protection without sacrificing data quality and users' enthusiasm. Specifically, the main contributions are, i) to enable privacy preservation and task allocation, we propose SPOON, a strong privacy-preserving mobile crowdsensing scheme supporting accurate task

allocation. In SPOON, the service provider recruits mobile users based on their locations, and selects proper sensing reports according to their trust levels without invading user privacy. By utilizing the blind signature, sensing tasks are protected and reports are anonymized. In addition, a privacy-preserving credit management mechanism is introduced to achieve decentralized trust management and secure credit proof for mobile users; ii) to improve communication efficiency while guaranteeing data confidentiality, we propose a fog-assisted secure data deduplication scheme, in which a BLS-oblivious pseudo-random function is developed to enable fog nodes to detect and delete replicate data in sensing reports without exposing the content of reports. Considering the privacy leakages of mobile users who report the same data, the blind signature is utilized to hide users' identities, and chameleon hash function is leveraged to achieve contribution claim and reward retrieval for anonymous greedy mobile users; iii) to achieve data statistics with privacy preservation, we propose a privacy-preserving data statistics scheme to achieve end-to-end security and integrity protection, while enabling the aggregation of the collected data from multiple sources. The correctness verification is supported to prevent the corruption of the aggregate results during data transmission based on the homomorphic authenticator and the proxy re-signature. A privacy-preserving verifiable linear statistics mechanism is developed to realize the linear aggregation of multiple crowdsensed data from a same device and the verification on the correctness of aggregate results; and iv) to encourage mobile users to participating in sensing tasks, we propose a dual-anonymous reward distribution scheme to offer the incentive for mobile users and privacy protection for both customers and mobile users in MCS. Based on the dividable cash, a new reward sharing incentive mechanism is developed to encourage mobile users to participating in sensing tasks, and the randomization technique is leveraged to protect the identities of customers and mobile users during reward claim, distribution and deposit.

Acknowledgements

The past four years of my PhD research at the University of Waterloo are truly the most unique, precious and awarding time in my life. I would like to thank all the people who greatly support my PhD study. They are my supervisors, my thesis committee members, and 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, whose patient guidance, valuable suggestions and constant encouragement make me successfully complete this thesis. He not only helps me develop the academic skills, but also guides me to strive for excellence. What I appreciate the most of Professor Shen is his great support and understanding to me. I am really inspired by his dedication and enthusiasm to his work, his students and his family. I also gratefully acknowledge my co-supervisor, Professor Xiaodong Lin, for his great efforts to inspire my research ideas and help me acquire research achievements. In addition, I would like to thank Professor Vojislav Misic for serving as my thesis external examiner and sharing his insights on network security with me. I also appreciate the honorable members of my thesis committee, Professor Jun Liu, Professor Sagar Naik and Professor Liang-liang Xie. Their insightful comments have significantly affected the substance and presentation of my work.

Almost 1500 days and nights were spent together with my colleagues in the lab. Although 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 Rongxing Lu, Professor Xiaohui Liang, Professor Kuan Zhang, Professor Kan Yang, Professor Ning Lu, Professor Ning Zhang, Professor Haibo Zhou, Dr. Miao Wang, Dr. Ran Zhang, Dr. Nan Cheng, Dr. Jian Qiao, Nan Chen, Wenchao Xu, Dongxiao Liu, Cheng Huang, Professor Yong Yu, Professor Man Ho Au, Professor Haomiao Yang, Professor Qi Jiang, Professor Dajiang Chen, Yuanyuan He, Yuan Zhang, Hao Ren, Jialu Hao, Meng Li and Wenjuan Tang, 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 here. It does not mean that I have forgotten them or their help. It is a privilege for me to work and share life with so many bright, energetic and helpful people.

The thesis is dedicated to my parents. 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 fulfil my career goals and never disappoint them.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AES	Advanced Encryption Standard
BBS+	Boyen-Shacham Signature Plus
BGN	Boneh-Goh-Nissim
BIDH	Computational Bilinear Inverse Diffie-Hellman problem
BLS	Boneh-Lynn-Shacham
BLS-OPRF	BLS-Oblivious Pseudo-Random Function
CBC	Cipher Block Chaining
CDH	Computational Diffie-Hellman
CONF	Conference-Key Sharing
CSP	Crowdsensing Service Provider
CTR	Counter
DARD	Dual-anonymous Reward Distribution
DDH	Decisional Diffie-Hellman
DHI	Diffie-Hellman Inversion
DL	Discrete Logarithm
DP	Differential Privacy
DSA	Digital Signature Algorithm
DTRPP	Dynamic Trust Relationships Aware Data Privacy Protection
EDE	Encrypt Decrypt Encrypt
EPPA	Efficient and Privacy-Preserving Aggregation
Fo-MCS	Fog-assisted Mobile Crowdsensing Framework
Fo-SDD	Fog-assisted Secure Data Deduplication
GCM	Galois/Counter Mode
GPS	Global Position System

IV	Individual Value
LRSW	Lysyanskaya-Rivest-Sahai-Wolf
MCS	Mobile Crowdsensing
MLE	Message-Lock Encryption
OC	Operation Center
P^{2}SM	Privacy-Preserving Smart Metering
PEPPeR	Privacy-Preserving Access Control Mechanism
PEPSI	Privacy-Enhanced Participatory Sensing Infrastructure
PPMS	P rivacy Preserving Mobile Sensing
PS	Pointcheval-Sanders
q**-DBDHI**	q-Decisional Bilinear Diffie-Hellman Inversion
q**-SDH**	q-Strong Diffie-Hellman
QoI	Quality-of-Information
RAPPOR	Randomized Aggregatable Privacy-Preserving Ordinal Response
SC-server	Spatial Crowdsensing server
SPK	Signature Proof-of-Knowledge
SPOON	Strong Privacy-preserving Mobile Crowdsensing
SRBE	Sublinear Revocation with Backward Unlinkability and Exculpability
SSL	Secure Sockets Layer
TA	Trusted Authority
TLS	Transport Layer Security
TPM	Trusted Platform Module
TTP	Trusted Third Party
VANET	Vehicular Ad Hoc Network
VPN	Virtual Private Network
ZKPoK	Zero-knowledge Proof-of-Knowledge

Chapter 1

Introduction

The integration of sensors and embedded computing devices triggers the emergence of mobile crowdsensing services, which allow individuals to cooperatively collect and share data and extract information to measure and map phenomena of common interest using sensing and communication devices. With the rapid development and increasing popularity of mobile devices, mobile crowdsensing becomes a broad range of sensing paradigms nowadays. Carrying mobile devices, users enable to collect, produce and upload various types of information to crowdsensing service providers via the Internet, ranging from general information, e.g., location, temperature and pollution level, to more specialized data, e.g., traffic awareness, driving behaviors, health condition and voting intentions. This new service has the potential for enormous social and economic impacts.

Despite tremendously wide applications, mobile crowdsensing is confronted with a variety of serious security and privacy threats, such as denial-of-service attacks, impersonation attacks and Sybil attacks. Moreover, sensitive information of mobile users may be obliviously disclosed to untrusted parties. However, we are in a dilemma when dealing with security and privacy issues in mobile crowdsensing. On one hand, if the security and privacy issues in mobile crowdsensing are not resolved in a satisfactory way, mobile users are disappointed to mobile crowdsensing services and stop participating in crowdsensing activities. On the other hand, if the personal information of mobile users is solidly preserved, the service provider is difficult to recruit appropriate mobile users to fulfill crowdsensing tasks based on their personal profiles. In this thesis, we investigate the security and privacy challenges and propose advanced secure and privacy-preserving mechanisms in order to resolve the challenges on security and privacy for mobile users, while ensuring key components of MCS efficient and effective, including task allocation, data collection, data analysis and reward feedback.

1.1 Mobile Crowdsensing

Mobile CrowdSensing (MCS) [1] is a large-scale sensing paradigm relying on the power of the crowds of mobile users. They cooperatively collect and share data and extract information to measure and map phenomena of common interest using their mobile devices. With the increasing number of mobile users sharing local knowledge (e.g., local events, noise level, ambient content and traffic conditions) acquired by their sensor-enhanced devices, the information is gathered in the cloud for large-scale sensing and community intelligence discovery. The mobility of large-scale mobile users makes MCS more versatile and flexible, such that data sensing infrastructure can be enriched. The formal definition of MCS is as follows: *a new sensing paradigm that empowers ordinary mobile users to contribute data sensed and collected from their sensor-enhanced devices, and aggregates and fuses the data in the cloud for crowd intelligence extraction and human-centric service delivery* [2]. From data sensing perspective, the success of MCS depends on a distributed data collection model, which is based on a general phenomenon (mentioned in a book titled *The Wisdom of Crowds* [3]) that the aggregation of data or information from a group of people often results in better decisions than those made by a single person from the group. The intelligence of a crowd is determined by four key qualities, namely, diversity in opinion, independence of thinking, decentralization and opinion aggregation. Through crowd-powered data collection, MCS can significantly improve the quality and credibility of sensing results. A broad range of applications are thus enabled, including traffic planning, mobile social recommendation, environment monitoring, and public safety. Therefore, increasing interests have been raised by both industry and academia on various MCS applications, aiming to increase the number of participating mobile users and improve the quality of crowdsensing results.

1.1.1 MCS Architecture

MCS architecture consists of three entities: service providers, customers, and mobile users.

Service Providers - Service providers usually develop cloud services by themselves or rent the cloud resources offered by cloud vendors. The service providers have sufficient storage and computing resources to offer MCS services. The service providers keep necessary information about customers and mobile users, such as identities, registration information and reputations. They receive crowdsensing tasks from customers and allocate them to mobile users. They also collect sensing reports from mobile users and generate crowdsensing results for customers. Besides, they are responsible for distributing rewards to the mobile users who make contributions on crowdsensing tasks.

Customers - The customers can be individuals, corporations or organizations. They need to accomplish data collection tasks, e.g., to study traffic congestion in a city, pollution level of a creek and satisfactory on public transportation, but they do not have sufficient capabilities to perform tasks by themselves, and thereby they turn to service providers for help. Specifically, customers send their crowdsensing tasks to the service providers and offer rewards to the participating mobile users. After the tasks are fulfilled, they can obtain the crowdsensing results from service providers.

Mobile Users - Every mobile user has several mobile devices, e.g., mobilephones, tablets, vehicles and items embedded with sensors, such as GPS, camera, proximity and microphone. These mobile devices, with rich computation, communication and storage resources, are carried by their owners wherever they go and whatever they do. The mobile users make sure the battery on mobile devices has sufficient power to support their normal functionalities. The mobile users can participate in crowdsensing tasks and utilize their portable devices to collect data from the environment based on the requirements of crowdsensing tasks, and report sensing data to the service providers. In addition, the mobile users would acquire some rewards from the service providers as benefits.

MCS consists of four phases, namely, task allocation, data collection, data analysis and reward feedback, as illustrated in Fig. 1.1.

- ▷ Task Allocation. A customer crowdsources a crowdsensing task to the service provider, along with the claimed rewards for attracting mobile users and other information used to evaluate the task fulfillment. The service provider accepts the task and allocates it to the mobile users according to the task requirements and the profiles of mobile users.

- ▷ Data Collection. Upon receiving the task, mobile users firstly determine whether accept the task or not. If yes, they start to perform the task by collecting data from their surrounding areas using the on-board sensors and pre-processing them to produce crowdsensing reports based on the task demands. Finally, they deliver the reports to the service provider.

- ▷ Data Analysis. When the service provider receives sufficient sensing reports from mobile users, it analyzes the sensing reports by performing several operations, e.g., truth discovery, data statistics and machine learning, and produce a crowdsensing result for the customer. The customer reads the crowdsensing result to obtain the knowledge and accomplish the task.

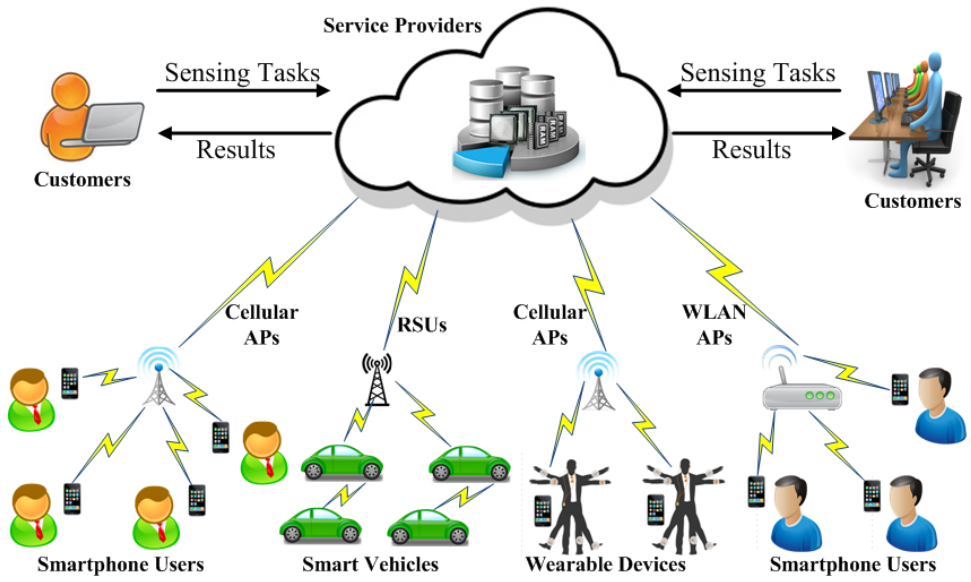


Figure 1.1: Mobile Crowdsensing Architecture

▷ Reward Feedback. The customer gives a feedback about the quality of the sensing reports, and the service provider distributes the rewards to the mobile users according to the feedback from the customer.

1.1.2 MCS Characteristics

MCS is a special case of wireless sensor networks, where mobile users carrying sensor-enhanced devices to collect data about common phenomena for some specific tasks voluntarily or for benefits. Therefore, MCS has unique characteristics that differentiate it from traditional sensor networks.

1. Rich Resources. In traditional sensor networks, limited computation capability and low battery storage issues of sensors are the major obstacles for the deployment of large-scale data collection. On the contrary, current mobile devices, such as mobilephones, tablets and in-vehicle sensing devices, have powerful computation and communication capabilities to reach the requirements of data collection and reporting, and sufficient energy level under the control of mobile users to make sure their normal functionalities, thereby enable various applications that require resources and sensing modalities [1].

2. Cost Saving. Millions of mobile devices have been deployed in the field. People carry these devices wherever they go and whatever they do. By utilizing these devices, it is possible for service providers to build large-scale sensing applications efficiently. For example, instead of installing roadside cameras and loop detectors, mobilephones carried by drivers can be used to collect traffic data and detect congestion levels.

3. High Scalability. MCS applications build on the data collected by a large volume of mobile users, the scalability is a basic feature of underlying communication systems. The communication protocols and network infrastructure are high distributed and decentralized, such that the participating mobile users have their freedom and convenience to join crowdsensing activities. The population of mobile users performing crowdsensing tasks can arbitrary fluctuate, even reaching an inconceivable number. Moreover, with the inherent mobility of users, the spatio-temporal coverage of crowdsensing applications is unprecedented compared with static sensor networks. The sensing area can be dramatically expanded according to the requirements of large-scale applications.

4. Human Involvement. Humans are usually involved in MCS applications, because the devices are owned and carried by individual users. They can arbitrarily participate in or stop MCS activities based on their preferences. The intelligence and mobility of users can be leveraged to collect high-quality and semantically complex data that might otherwise require sophisticated software and hardware in traditional sensor networks [4]. For instance, individuals can easily identify accessible street parking spots and report photos or messages for providing parking navigation services, whereas an ultrasound-based scanning system should be developed, which not only needs special hardware, e.g., camera, but also processing algorithms to identify vacant parking places.

1.1.3 MCS Applications

MCS applications can be classified into three major categories based on the type of phenomena being measured or mapped, namely, environmental, infrastructure and social [1].

1.1.3.1 Environmental Applications

In environmental MCS applications, the phenomena are those of national environment. These applications include measuring air pollution level in a city, noise level in a residential area, water level in creeks, and monitoring wildlife habitats and vegetation protection. In general, individuals usually participate in tasks of mapping various phenomena of large-scale environment voluntarily for the purpose of environmental protection.

Nature Preservation. During the past few years, mobile users utilized their devices to contribute data for scientific studies. For example, Great Backyard Bird Count project counts the wild birds in the United State with the help of volunteers; MIT Owl project studies owl population by leveraging the network of sensor-powered smartphones. Besides, to study the impacts of climate change, scientists employ citizens to collect some specific data related to their research topics, such as the link between increasing temperatures and the timing of plants events (e.g., emergence of first leaf, flowering and fruiting) [5]. Researchers can release some tasks to recruit mobile users to collect data about soil erosion, deforestation, melting glaciers, etc. The prospect of large-scale information collection for nature preservation with the involvement of mobile users is becoming a reality.

Pollution Measurement. Environmental pollution has become a worldwide problem, including water pollution, noise pollution, air pollution and light pollution. Pollution sources discovery and pollutants control need the extensive participation of citizens. For instance, European Commission mandates the creation of noise contour maps to gather information about exposure. However, government efforts are limited since in most cases it is impossible to deploy sensing nodes to cover all areas of a city for data collection. This issue can be remedied through the efforts of all mobile users, who can use the microphones in smartphones to measure the ambient noise level. These data aggregated from the volunteers in the city can be used to generate a fine-grained noise map. UAir [6] gathers fine-grained air pollution by utilizing heterogeneous crowdsensed data, measured from sensing stations, traffic information, points of interest, etc.

1.1.3.2 Infrastructure Applications

Infrastructure MCS applications involve the measurements of large-scale phenomena related to public infrastructure, including measuring parking availability [7], queuing time in hospitals, traffic congestion, real-time transit tracking, road conditions and outages of public facilities (e.g., malfunctioning fire hydrant, confusing road sign and broken traffic lights) [8].

Traffic Condition Monitoring. Traffic jams have negative social and economic effect on society, bringing time-consuming and frustrating experiences to drivers and leading to critical social problems, e.g., fuel waste, air pollution and accidents [9]. By using data from GPS-equipped vehicles and mobilephones, a vehicle can avoid itself being congested on road and find a proper route to reach its desirable destination with low traffic delay and fuel cost. For instance, Google Map employs real-time data contributed by hundreds of millions of mobilephones around the world for traffic and road condition analysis. Google can analyze the total number of cars and their speed on a road at any given time using

anonymous bits of data reported by all mobilephones with Google Maps applications. Furthermore, a combination of traffic data from cars, taxies and smart card records from buses can be used to study human traffic at a hotspot, which can also provide important clew for public transportation design.

Parking Navigation. With the increasing number of vehicles in metropolises, parking in a congested area, such as downtown and shopping mall, particularly in peak hours, has been a conflicting and confusing problem for a large number of drivers [10]. It is common for drivers to circulate on roads in a congested region, looking for accessible parking spaces. These vehicles may cause an average 30% of the traffic on roads. These extra traffic also leads to serious social problems, such as fuel waste, traffic congestions, air pollution and vehicle accidents. Real-time parking information can assist drivers to find available parking spaces quickly. Nevertheless, it is pretty different to collect and publish the parking information, particularly for the roadside parking information. The video camera on vehicles can record the driving scene, from which the cloud can acquire the information about vacant parking spaces on the streets and in the parking lots. Therefore, the moving vehicles can upload the driving video or photos to the fog nodes and a vehicle who looks for a parking space can send a parking query to the cloud, including its destination, arriving time, expected price, etc. [11]. The cloud can retrieve and analyze the video and photos from the fog node covering the destination to find a vacant parking space for the querying vehicle.

Road Surface Monitoring. The detection of road surface abnormities (e.g., potholes, bumps, ice, railway crossing) and their locations contribute to the improvement of road condition and drivers' safety [12]. Road quality assessment has been identified as an important issue related to the possibility of making the drivers and passengers more comfortable, safe and efficient. Nericell [13] could detect and report road conditions using the built-in sensors in mobile devices. These data are further integrated with the traffic maps to share with the public. The transportation agencies or municipalities can automatically recognize the road surface abnormities in the region of their jurisdiction for prioritizing road repairing.

1.1.3.3 Social Applications

In social MCS applications, individuals share sensing data (e.g., traveling experiences) or facilities (vehicles, bicycles and umbrella) with others [14]. As an example, individuals share their experiences and recommend them to the rest of the community, and thereby improving the experiences of others. We give two social MCS applications as examples, namely, social recommendation and ride sharing.

Social Recommendation. Social recommendation applications enable customers to find their interested services based on the wealth of data reported by mobile users. These applications provide personalized recommendation by exploring mobile crowdsensing data, and allow customers to set their own planning to fulfill tasks. Place recommendation is one of important applications in MCS, e.g., restaurant recommendation, park recommendation and hotel recommendation. It leverages the historical location trajectories recorded by mobile devices for recommendation. GeoLife [15] measures the similarity of customers according to the location history to offer personalized place recommendation services. Itinerary planning can recommend travel routes to tourists based on individual constraints, such as time schedule, expenditure budget and user preference.

Ride Sharing. Ride sharing services [16] provide partner discovery to drivers and riders with similar rides for initializing sharing travel experiences. A driver sends a ride offer or a rider sends a ride request to a ride-sharing server, and the server helps the driver to find rider-share partners with similar itineraries. Ride-sharing services allow drivers to share vacant seats in their vehicles on the road, bringing various benefits to individual users, e.g., improved vehicle occupancy, shared travel costs and extended social circles, and the society, e.g., reduced traffic congestion, fuel consumption and carbon dioxide emissions. Many service providers have emerged to offer ride-share partner discovery services, e.g. Flinc, Lyft Line, UberPool, Waze Carpool and Blablacar. Ride-sharing has become increasing popular in metropolis to reduce crowded traffic and expensive transportation costs.

1.2 Security and Privacy in MCS

The sensing reports collected by on-board sensors carried in mobile devices, are conceptually tied to specific mobile users and thereby infer personal information and activities about the users. Once the data are uploaded to the cloud, the mobile users lose physical control over their collected data. The MCS service providers have their own incentives and the servers are vulnerable to be compromised or attacked. The corruption of sensing reports may directly impact the trustworthiness of crowdsensing results, and further mislead the customers to make irrational decisions. Therefore, data protection and privacy preservation are significantly important for both customers and mobile users.

1.2.1 Security and Privacy Threats

The service provider may not be fully trusted, it has numerous motivations to share the detailed crowdsensed data with their cooperators for monetary reasons. For example, to

promote the new platform, Uber Movement, Uber has released staggering 2 billion pieces of trip data collected from people in more than 450 cities. Further, data exposure accidents frequently happen on the data centers of corporations, such as Facebook, Yahoo and Apple, dramatically reduce users' trust in application providers. In addition to insider threats from service providers, MCS may face with a variety of outsider attacks, such as eavesdropping attack, forgery attack, impersonation attack, spam attack, Sybil attack, collusion attack and man-in-the-middle attack.

- ▷ Eavesdropping: Malicious attackers listen on communication channels to capture transmitting crowdsensed reports and read the captured data.
- ▷ Forgery: Malicious attackers may not only forge their identities and profiles, but also generate fake crowdsensed data to mislead customers.
- ▷ Impersonation: A malicious attacker pretends a legitimate user to enjoy the MCS services, or impersonates a legitimate mobile user to provide data to mislead customers.
- ▷ Spam: Spam data refer to the unwanted content, such as redundant information, false collected data from users, which are generated and spread by attackers.
- ▷ Sybil: Sybil attackers either manipulate fake identities or abuse pseudonyms in order to compromise or control the effectiveness of mobile crowdsensing. For example, they could generate incorrect crowdsensing reports, such that the crowdsensing results may not be trustworthy.
- ▷ Collusion: Two or more mobile users collude together to deceive, mislead, or defraud service providers and customers.
- ▷ Man-in-the-Middle: A malicious attacker stands in the middle of two parties to secretly relay or modify the exchanging data between these parties, however, these two parties believe that they are directly communicating with each other.

Further, both customers and mobile users have the incentives to behave greedily on rewards. Specifically, customers may refuse to pay the rewards e.g., money, coupons, and services, they claimed before, and mobile users strive to obtain more benefits from task participation, but are unwilling to pay equal efforts on task fulfillment. They may launch double-reporting attack and double-claiming attack, in which they report the same crowdsensed data or claim a reward more than once, respectively.

Privacy is a critical issue as the users' sensitive data are involved in the collection, transmission, processing and sharing in MCS. Data owners are unwilling to expose their privacy to others, but the leakage of privacy is oblivious in task allocation and data collection. A user's privacy may include three aspects, that is, identity privacy, data privacy, and location privacy.

- ▷ Identity Privacy. The identity of a user includes the name, address, telephone number, visa number, license number and public-key certificate that any information can be linked to a specific user.
- ▷ Data Privacy Crowdsensed data are reported to the service provider, who is not honest, ends up with the privacy leakage for mobile users. Further, the crowdsensed tasks are exposed to both the service provider and mobile users, which may result in the privacy corruption of customers, since the motivation of releasing the tasks can be predicated based on the task contents.
- ▷ Location Privacy. Massive MCS applications require the mobile users to collect data at specific areas, such that the service provider should allocate these tasks to the mobile users near or in the areas. The location feature in MCS is important for task allocation and task fulfillment, so as to expose the locations or trajectories of both customers and mobile users.

1.2.2 Security and Privacy Requirements

To secure the MCS applications against the aforementioned threats, security and privacy requirements should be reached to promote the healthy development.

1.2.2.1 Security

Confidentiality, authentication and integrity should be achieved to ensure the security of MCS applications.

Confidentiality: MCS applications entail serious security threats. Firstly, data collected by mobile users encapsulates various aspects of physical environment, including social events, pollution levels, traffic conditions and personal activities. Some data may be considered sensitive, e.g., personal activities, health status and personal information about individuals, while others are not, e.g., pollution levels and social events. Therefore, it is critical for mobile users to distinguish sensitive information from large volumes of data

before uploading. Whether the data are sensitive or not is totally determined by the user, and each has his/her personal preference and choice. If the sensing data with sensitive information are not well preserved, the curious entities, such as service providers and mobile users, are able to extract various sensitive information from the sensing reports. Therefore, protecting the confidentiality of sensing data is the primary objective.

Authentication: Authentication is a critical aspect related to the functionality of sensing reports, which contribute to fulfill the tasks for customers. If these reports are delivered by untrusted or malicious mobile users, the customers may be confused by the results and make false decisions, so that it is worthwhile to assure that the sources of sensing reports are fully-trusted and behave honestly. In addition, the lack of authentication on customers brings troubles on task releasing. For example, the attackers may crowdsource invalid tasks to the service provider spitefully and capture the crowdsensing results released by honest customers to enjoy free crowdsensing services. Therefore, it is necessary to guarantee that only the honest customers and mobile users can participate in MCS activities.

Integrity: It is of critical importance to preserve the integrity of crowdsensing reports when they are maintained on cloud. Due to the limited storage space, the cloud may maliciously discard the unexpired sensing reports to mislead customers. For example, an attacker may compromise the cloud and corrupt the video of a traffic accident to escape the punishment. Not only can attackers modify or forge the sensing reports, but also corrupt data processing to generate biased results to impact the customers' decision. Therefore, how to ensure the trustworthiness of crowdsensing results become essential. The results should be unbiased and uncontrolled by malicious attackers. The correctness verification of results is significantly essential from the perspective of customers.

1.2.2.2 Privacy

Both mobile users and customers may concern their privacy leakage in MCS.

Privacy of Mobile Users: The sensors on mobile devices collect the data from environment. These data are necessarily people-centric and related to some aspects of mobile users and their social setting: where mobile users are and where they are going; what places they are frequently visited and what they are seeing; how their health status is and which activity they prefer to do. For example, a mobile user Alice may want to report a traffic jam in downtown, without the service provider knowing that Alice is congested in downtown at the time she reports the event. Therefore, protecting the location of mobile users when they are reporting data is critical in MCS applications. Although anonymity becomes significantly important for MCS applications, once the sensing reports are kept

anonymous, no one can identify the contributors of reports, such that it is difficult for the service provider to distribute the benefits to the corresponding mobile users according to their distinct contributions on tasks.

Privacy of Customers: Customers are willing to release their crowdsensing tasks without exposing their identities, since these tasks may also contain some sensitive information, from which the curious service provider can predict the reasons why customers issue these tasks. How to enable the service provider to recruit mobile users for sensitive tasks is essential. One trade-off is to expose the contents of tasks, but protecting the identities from being known instead. This scarification might be acceptable for customers since the service provider cannot link the identities of customers with the content of tasks effectively. Nonetheless, anonymous customers may escape the payments of rewards to mobile users. Therefore, it is necessary to design reward claiming and distribution schemes to achieve anonymous payments for customers and mobile users undeniably and efficiently.

1.2.2.3 Fairness

Although MCS applications are designed as best-effort services, in which mobile users voluntarily participate in data sensing and reporting, these operations would cost storage, bandwidth and battery of mobile devices and sacrifice partial privacy for mobile users. These issues may degrade the enthusiasm of mobile users for task participation. One major challenge is to encourage mobile users to report real-time information, especially if a threat to their privacy. The best approach is to provide sustainable incentive. With direct and indirect benefits for mobile users, they are easy to make contributions on data collection. However, the fairness is emerging as a new challenge to balance, which includes two aspects: customers' fairness and drivers' fairness.

Fairness of Customers: The crowdsensing results acquired by customers should deserve the costs they paid. The participating mobile users may be greedy on benefits and lazy for sensing. On one hand, mobile users make their best effect to offer better crowdsensing reports for earning benefits. On the other hand, mobile users have an incentive to cheat, to obtain more rewards than they fairly deserve. For example, drivers may use multiple identities in disguise to report false traffic information. The misbehavior of mobile users can lead to the unfairness for customers, because their acquirements do not match the cost they paid due to the untrustworthy crowdsensing results. As a result, customers may be disappointed in MCS services, directly impacting its fully flourish.

Fairness of Mobile Users: Fairness of mobile users means that the mobile users should be rewarded their deserved benefits according to their contributions on data collection.

In reward distribution, the customers determine the number of benefits that a mobile user should be rewarded, and the service provider is responsible for assigning the benefits to mobile users. During these processes, how many benefits the mobile users obtain is absolutely controlled by the customers and the service providers. Thus, the mobile users may be rewarded less benefits than they fairly deserve, because of the renege of customers and the embezzlement of service providers. Specifically, the customers may refuse to pay the benefits or just fulfill partial benefits they promised in task releasing, and the service provider embezzles part of benefits and only assigns the rest to the participating mobile users. This misbehavior seriously damages the enthusiasm of mobile users.

1.3 Research Challenges and Objectives

The objective of this thesis is to develop a set of efficient, secure and privacy-preserving schemes to countermeasure and mitigate the aforementioned security and privacy threats. More importantly, the schemes should reach the security, privacy and fairness requirements for every step of MCS, from task allocation, data collection to data analysis and reward feedback. To reach these demands, we aim to address the following challenges:

- Privacy-preserving Task Allocation: To enhance data quality, the service provider is required to recruit mobile users based on their personal information, e.g., trajectory and reputation, which could end up with the privacy leakage of mobile users, unfortunately. On the other hand, the protection of users' information results in the difficulty on task allocation and data quality guarantees. Therefore, how to enable accurate task allocation, while preserving the privacy of mobile users is the first challenge in MCS.

- Secure Data Collection: The success of MCS largely depends on the participating mobile users. The broader participation, the more sensing data are collected; nevertheless, the more replicate data may be generated, thereby bringing unnecessary heavy communication overhead. Hence it is critical to eliminate duplicate data to improve communication efficiency, a.k.a., data deduplication. Unfortunately, sensing data are usually protected, making their deduplication challenging than ever. Therefore, how to enable data confidentiality and data deduplication in data reporting is the second challenge in MCS.

- Privacy-preserving Data Analysis: Crowdsensed data analysis is a key component of MCS, in which the service provider is enabled to produce crowdsensing results from

the collected data. To keep data confidentiality, how to support the crowdsensed data analysis without exposing the detailed data contents is of significant importance. Moreover, since the service provider and the intermediates during transmission are not fully trusted, the trustworthiness of the results becomes a serious concern for the customer. The corruption on the crowdsensed data and the crowdsensing results are difficult to be detected and prevented in MCS.

- Fair and Secure Reward Feedback: To encourage the participation of mobile users, rewards should be distributed to anonymous mobile users in reward feedback with privacy preservation. However, it is hard to ensure all mobile users honestly retrieve the rewards that they deserve to obtain, and fairly receive the coins that the customers claim to give. The cheaters, i.e., customers, mobile users and the service provider, have sufficient motivations to undermine the reward distribution for their own benefits.

1.4 Research Contributions

To achieve the above objectives, we develop a suite of schemes based on advanced security and privacy enhancing technologies in MCS. Specifically, the main contributions lie in the following aspects:

- *Strong Privacy-preserving Task Allocation*: A strong privacy-preserving mobile crowdsensing scheme (SPOON) is introduced to support accurate task allocation according to spatialtemporal information and credit levels of mobile users. In SPOON, the service provider enables to recruit mobile users based on their locations, and select proper sensing reports according to their trust levels without invading user privacy. By utilizing proxy re-encryption and BBS+ signature, sensing tasks are protected and reports are anonymized to prevent privacy leakage. In addition, a privacy-preserving credit management mechanism is introduced to achieve decentralized trust management and secure credit proof for mobile users. Finally, we show the security properties of SPOON and demonstrate its efficiency on computation and communication.

- *Secure Data Deduplication*: We propose a fog-assisted secure data deduplication scheme (Fo-SDD) to improve communication efficiency while guaranteeing data confidentiality. Specifically, a BLS-oblivious pseudo-random function is designed to enable fog nodes to detect and remove replicate data in sensing reports without exposing the contents of reports. To protect the privacy of mobile users, we further extend Fo-SDD

to hide users' identities during data collection. In doing so, Chameleon hash function is leveraged to achieve contribution claim and reward retrieval for anonymous mobile users. Finally, we demonstrate that both schemes achieve secure, efficient data deduplication.

- *Privacy-preserving Data Statistics*: We investigate privacy-preserving data statistics on crowdsensed meter readings from smart meters in smart grid. We define a new security model to formalize the misbehavior of collectors, in which the misbehaving collectors may launch pollution attacks to corrupt crowdsensed consumption data during transmission. Under this model, we propose a novel privacy-preserving data statistics scheme on the crowdsensed data collected by multiple smart meters to prevent pollution attacks, and a privacy-preserving verifiable linear statistics mechanism to realize the linear aggregation of multiple crowdsensed data and the verification on the correctness of aggregate results.

- *Dual-anonymous Reward Feedback*: We propose a dual-anonymous reward distribution scheme (DARD) to achieve the incentive for mobile users and privacy protection for both customers and mobile users in mobile crowdsensing. Specifically, we design a reward sharing incentive mechanism to encourage mobile users to participate in tasks and employ the randomizable technique to protect the identities of customers and mobile users during reward claim, distribution and deposit. Our analysis further shows that DARD achieves reward balance and cheater detection with low 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 related literatures in security and privacy of MCS. Chapter 3 develops a strong privacy-preserving task allocation scheme to achieve accurate allocation and privacy preservation of mobile users and customers. Chapter 4 investigates to achieve data confidentiality and data deduplication in MCS by proposing a fog-assisted secure data deduplication scheme for crowdsensed data protection and communication efficiency improvement. Chapter 5 proposes a privacy-preserving data statistics scheme on crowdsensed data from smart meters to ensure the end-to-end security and the correctness of the statistical results. Chapter 6 designs an efficient dual-anonymous reward distribution scheme to achieve reward-sharing incentive

and prevent privacy leakage for customers and mobile users. Finally, Chapter 7 concludes the thesis, and introduces our future research directions.

Chapter 2

Background

This chapter introduces the background of security and privacy for MCS. We first review the underlying techniques leveraged to design the proposed schemes. Then, we give a comprehensive survey on the literature of security and privacy in MCS.

2.1 Basic Techniques

We review the preliminaries, including the bilinear map, number-theoretic problems, BBS+ signature, PS signature, proxy re-encryption, zero-knowledge proof, and blockchain.

2.1.1 Bilinear Map

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

The bilinear map above is type 3 pairing [17], in which $\mathbb{G}_1 \neq \mathbb{G}_2$ and there is no efficiently computable homomorphism between $\mathbb{G}_1$ and $\mathbb{G}_2$ in either direction. Type 1 pairing is that

$\mathbb{G}_1 = \mathbb{G}_2$; and type 2 pairing is that $\mathbb{G}_1 \neq \mathbb{G}_2$ and there exists an efficiently computable homomorphism $\pi : \mathbb{G}_2 \to \mathbb{G}_1$, but there is no efficient homomorphism in the other direction, according to the definition due to Galbraith et al. [18].

2.1.2 Negligible Function

If S is a non-empty set, $s \in_R S$ denotes s is randomly chosen from S. We say that a function $g(\lambda)$ is a negligible function, if for every positive polynomial $f(x)$, there exists an integer $N > 0$ such that for all $x > N$, $g(x) < \frac{1}{f(x)}$.

2.1.3 Number-Theoretic Problems

The security of many cryptosystems relies on the intractability of solving some hard problems. We present the following problems that are relevant to this thesis. The respective assumptions state that no probabilistic, polynomial time algorithm has non-negligible advantage in solving the corresponding problems.

Computational Diffie-Hellman (CDH) assumption [19]. If there is no algorithm can solve the CDH problem, that is, given $(g, g^a, g^b) \in \mathbb{G}_1^3$, to compute g^{ab}, in probabilistic polynomial time with non-negligible probability, then we say that the CDH assumption in $\mathbb{G}_1$ holds.

Decisional Diffie-Hellman (DDH) assumption in $\mathbb{G}_2$ [19]. If there is no algorithm can solve the DDH problem, that is, given $(\hat{g}, \hat{g}^a, \hat{g}^b, \hat{g}^c) \in \mathbb{G}_2^4$, to determine $c = ab$ or not, in probabilistic polynomial time with non-negligible probability, then we say that the DDH assumption in $\mathbb{G}_2$ holds.

Decisional Diffie-Hellman (DDH) Assumption in $\mathbb{G}_T$ [20]. The DDH problem in $\mathbb{G}_T$ is defined as follows: Given a tuple $(D, D^a, D^b, D^c) \in \mathbb{G}_T^4$, output yes if $c = ab$ and no otherwise. We say that the DDH assumption in $\mathbb{G}_T$ holds if there is no algorithm can solve the DDH problem in $\mathbb{G}_T$ with non-negligible advantage in probabilistic polynomial time.

Conference-Key Sharing (DHI) Assumption [21]. The DHI problem is defined as follows: Given $g, g^s \in \mathbb{G}_1$, where $s \in \mathbb{Z}_p$, to compute $g^{\frac{1}{s}} \in \mathbb{G}_1$. We say that the DHI assumption holds if there is no algorithm can solve the DHI problem with non-negligible advantage in probabilistic polynomial time.

Decisional Diffie-Hellman (CONF) Assumption [20]. The CONF problem is defined as follows: Given $g, g^s, g^{sv} \in \mathbb{G}_1$, $\hat{g} \in \mathbb{G}_2$ where $s, v \in \mathbb{Z}_p$, to compute $\hat{e}(g, \hat{g})^v \in \mathbb{G}_T$. We say that the CONF assumption holds if there is no algorithm can solve the CONF problem with non-negligible advantage in probabilistic polynomial time.

Bilinear Inverse Diffie-Hellman (BIDH) Assumption [22]. The BIDH problem is defined as follows: Given a tuple $(g, g^a, g^b) \in \mathbb{G}_1$, $\hat{g} \in \mathbb{G}_2$, to compute $\hat{e}(g, \hat{g})^{c/a}$. We say that the BIDH assumption holds if there is no algorithm can solve the BIDH problem with non-negligible advantage in probabilistic polynomial time.

q-Strong Diffie-Hellman (q-SDH) Assumption [20]. The q-SDH problem in $\mathbb{G}$ 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 Bilinear Diffie-Hellman Inversion (q-DBDHI) Assumption [22]. The q-DBDHI problem is defined as follows: For random $g \in \mathbb{G}_1$, $\hat{g} \in \mathbb{G}_2$ $x \in \mathbb{Z}_q$, $Q \in \mathbb{G}_T$, given $(g, g^x, g^{x_2}, \cdots, g^{x^q}, Q)$, to decide if $Q = \hat{e}(g, \hat{g})^{1/x}$ or not. We say that the q-DBDHI assumption holds if there is no algorithm can solve the q-DBDHI problem with non-negligible advantage in probabilistic polynomial time.

Modified LRSW assumption 1 [17]. If there is no algorithm can solve the modified LRSW problem 1, that is, given g^b, $\hat{g}^a$, $\hat{g}^b$, where g is a generator of $\mathbb{G}_1$, $\hat{g}$ is a generator of $\mathbb{G}_2$ and $a, b \in_R \mathbb{Z}_p$, and an oracle $\mathcal{O}$, which on input $m \in_R \mathbb{Z}_p$ that chooses a random $h \in_R \mathbb{G}_1 \backslash 1_{\mathbb{G}_1}$ and answers the pair $P = (h, h^{a+bm})$, to compute a new pair $P' = (h', h'^{a+bm'})$ for $h' \in_R \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and a new m' that is not one of the ms queried in $\mathcal{O}$, in probabilistic polynomial time with non-negligible probability, then we say that the modified LRSW assumption 1 holds.

Modified LRSW assumption 2 [17]. If there is no algorithm can solve the modified LRSW problem 2, that is, given $\hat{g}^a$, $\hat{g}^b$, where $\hat{g}$ is a generator of $\mathbb{G}_2$ and $a, b \in_R \mathbb{Z}_p$, and an oracle $\mathcal{O}$, which on input $m \in_R \mathbb{Z}_p$ that chooses a random $h \in_R \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and answers the pair $P = (h, h^{a+bm})$, to compute a new pair $P' = (h', h'^{a+bm'})$ for $h' \in_R \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and a new m' that is not one of the ms queried in $\mathcal{O}$, in probabilistic polynomial time with non-negligible probability, then we say that the modified LRSW assumption 2 holds.

The modified LRSW assumption 1 and the modified LRSW assumption 2 can be proved to hold in the generic group model.

2.1.4 BBS+ and PS Signatures

BBS+ Signature [23]. Here we briefly review the BBS+ signature due to [23], which can be utilized to sign ℓ-message vector $(m_1, \cdots, m_\ell)$.

Let $g, g_1, \cdots, g_{\ell+1}$ be generators of $\mathbb{G}$. Randomly choose x from $\mathbb{Z}_p$ as the secret key of the signature scheme, and compute the corresponding public key as $y = g^x$.

A signature on messages $(m_1, \cdots, m_\ell)$ is (A, e, s), where $A = (g g_1^{m_1} \cdots g_\ell^{m_\ell} g_{\ell+1}^{s})^{\frac{1}{x+e}}$ and (e, s) are random values chosen from $\mathbb{Z}_p$.

This signature can be checked as: $\hat{e}(g g_1^{m_1} \cdots g_\ell^{m_\ell} g_{\ell+1}^{s}, g) \stackrel{?}{=} \hat{e}(A, y g^e)$.

The security of BBS+ signature can be reduced to the q-SDH assumption and it can be utilized to construct a zero-knowledge proof-of-knowledge protocol that allows the signer to prove the possession of the message-signature pair.

PS Signature [17]. The PS signature is a public-key signature scheme proposed by Pointcheval and Sanders [17] and its existential unforgeability is proven against chosen message attacks without random oracles under the modified LRSW assumption 2 [17].

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. 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 $\mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$. The signature ϕ can be publicly verified as $\phi_1 \neq \mathbb{G}_1 \setminus 1_{\mathbb{G}_1}$ and $\hat{e}(\phi_1, \widehat{Y} \prod_{j=1}^{r} \widehat{X}_j^{m_j}) = \hat{e}(\phi_2, \hat{g})$.

2.1.5 Proxy Re-encryption

Proxy Re-Encryption [22]. Proxy Re-encryption is a special public key encryption with a desirable property that a semi-trusted proxy enables to convert a ciphertext for Alice into a ciphertext for Bob without seeing the underlying plaintext, given a proxy re-encryption key. Thanks to this promising property, it has been widely employed in data sharing scenarios. The proxy re-encryption scheme is proposed by Ateniese et al. [22], the details of which are as follows:

- ▷ KeyGen($\cdot$). Alice picks a random value $a \in \mathbb{Z}_p$ as the secret key sk_a and compute the public key $pk_a = g^a$.
- ▷ RKeyGen(sk_a, pk_b). Alice delegates to Bob by sending the re-encryption key $rk_{A \to B} = g^{b/a}$ to a proxy by using Bob's public key.
- ▷ Encrypt(m, pk_a). To encrypt a message $m \in \mathbb{G}_T$ under pk_a, Alice chooses a random value $k \in \mathbb{Z}_p$ to compute $c_a = (g^{ak}, m\hat{e}(g, g)^k)$.

- ▷ Re-Enc$(c_a, rk_{A\to B})$. The proxy can change the ciphertext c_a into a ciphertext c_b for Bob with $rk_{A\to B}$. From c_a, the proxy calculates $\hat{e}(g^{ak}, g^{b/a}) = \hat{e}(g,g)^{bk}$ and releases $c_b = (\hat{e}(g,g)^{bk}, m\hat{e}(g,g)^k)$.
- ▷ Decrypt (c_b, sk_b). Bob enables to decrypt c_b to obtain m as $m = m\hat{e}(g,g)^k/(\hat{e}(g,g)^{bk})^{1/b}$.

2.1.6 Zero-Knowledge Proof

In a proof-of-knowledge protocol [24], a prover convinces a verifier that he knows a witness w satisfying some kind of relation R with respect to a known string x. That is, the prover can convince the verifier that he knows some w satisfies the relation $(w, x) \in R$. If the prover can convince the verifier in a way that the latter cannot learn anything except the validity of the relation, this protocol is called a zero-knowledge proof-of-knowledge (ZKPoK) protocol [25]. Currently, a plethora of ZKPoK protocols have been proposed, in which Σ-protocols are a special type of three-move ZKPoK protocol. They can be transformed into non-interactive Signature Proof-of-Knowledge (SPK) protocols or signature schemes that can be proven secure in random oracle model. Σ-protocols are able to be converted into 4-move perfect zero-knowledge proof-of-knowledge protocols [26].

For instance, $PK\{(x) : y = g^x\}$ denotes a Σ-protocol that proves the knowledge of discrete logarithm. That is, a prover convinces a verifier that he possesses the knowledge of $x \in \mathbb{Z}_p$ such that $y = g^x$ with respect to some $y \in \mathbb{G}$ without exposing the actual value of x. The values on the left of the colon denote the knowledge that the prover aims to prove, and the values on the right of the colon denote the publicly known values. The signature of knowledge for message $m \in \{0,1\}^*$ that is transformed from the above Σ-protocol is denoted as $SPK\{(x) : y = g^x\}(m)$, which is secure under the random oracle model due to Fiat-Shamir heuristic.

2.1.7 Blockchain

A blockchain [27] is a linear collection of data elements, where each data element is called block. All blocks are linked to form a chain and secured using a cryptographic hash function. Each block typically contains a hash pointer as a link to a previous block, a timestamp, and transaction data. Only if a transaction's validity is verified, it can be recorded into the block. Generally, the blockchain technique can be classified into two types: private blockchain and public blockchain [28]. For a private blockchain (including the consortium blockchain), the verification is performed by authorized participants, who

may be employed by the blockchain managers or the managers themselves. For a public blockchain, the verification can be performed by any participant in the network: a transaction can be recorded into a block, only if it has been verified and accepted by a considerable majority [29].

The most prominent manifestation of public blockchain is blockchain-based currencies (i.e., on-chain currencies), such as Bitcoin and Ethereum. In these currencies, the public blockchain is used to serve as an open and distributed ledger that efficiently records transactions between two participants. Furthermore, such ledger is verifiable and inherently resistant to modification of chained blocks. The participants who perform the transaction verifications and maintain the blockchain are called miners. Since the Ethereum is more expressive than other on-chain currencies, the ledger of Ethereum can be thought as a state transition system, where there is a "state" consisting of the ownership status of all existing Ethers (which are the value token of the Ethereum blockchain) and a"state transition function" that takes a state and a transaction as input, and outputs a new state as the result. When a new block is added into the chain, all transactions recorded in the block should be verified, and then miners compute a valid nonce such that the hash value of the block is less than or equal to a value provided by the Ethereum system [30]. The first miner who finds the nonce broadcasts the block of transactions together with this nonce. Other participants can verify that the nonce is a valid solution, and hence add the new block to their blockchain. Once the block is added to the chain, all the corresponding state information has been updated.

2.2 Related Work

We comprehensively review the literature about security and privacy in MCS, which are divided into five categories based on their different functions in MCS.

2.2.1 Privacy Protection for Mobile Users and Customers

In MCS, communications between mobile devices and the service provider depend on typical wireless and wired connections, e.g., cellular, WiFi, cable television, Internet access, and fiber-optic communication, and there are classical secret communication protocols and standards to support secure data transmission, such as secure communications interoperability protocol, secure shell, textsecure protocol, SSL/TLS, secure electronic transaction and Virtual Private Network (VPN). It is still challenging to provide security assurance

and privacy preservation on crowdsensing tasks and reports, since the architecture of MCS is complicated and all entities in MCS may be curious or malicious. Security and privacy issues in MCS have attracted extensive attentions from both academia and industry since the concept is proposed by Ganti et al. [1] in 2011. Yang et al. [31] defined security and privacy issues in MCS by identifying the sensing modalities and assessing the threats to user's privacy, and suggested to utilize several cryptographical methodologies to achieve privacy preservation. To offer solid security and privacy guarantees for mobile users, Cornelius et al. [32] proposed AnonySense, a basic architecture for privacy-aware task releasing and data sensing. The participating mobile devices collaboratively produce data and submit them through Mix networks. To achieve user privacy and data trustworthiness, Huang et al. [33] demonstrated that mobile users are vulnerable to linking attack if they naively reveal their reputations to the service provider and presented an anonymization scheme from pseudonyms and a reputation management mechanism by employing a trusted server to minimize the risk of such attack. Korshunov et al. [34] proposed a subjective evaluation mechanism using crowdsourcing to analyze the tradeoff between privacy preservation of mobile users and intelligibility of activities under video surveillance. Apart from the privacy of mobile users, Dimitriou et al. [35] raised the problem of customer's privacy leakage and proposed a privacy-preserving access control mechanism (PEPPeR) in sensing applications, which focuses on the privacy preservation for customers. Specifically, PEPPeR allows customers to obtain tokens from the service provider in order to have access to the data provided by the participating mobile users. Christin et al. [36] presented IncogniSense, an anonymous reputation framework, to preserve the identity privacy of mobile users. The pseudonyms are generated from blind signatures, such that their identities cannot be linked in multiple time periods. IncogniSense relies on a secure reputation transfer mechanism between pseudonyms, in which the reputations can be transferred for mobile users associated with adjacent time periods. Kazemi and Shahabi [37] defined privacy and trust in crowdsensing systems and proposed a trustworthy and privacy-preserving framework to increase the validity of collected data by recruiting multiple participants at each sensing location redundantly.

However, none of above frameworks enables to preserve privacy for both mobile users and customers. Therefore, Cristofaro and Soriente [38] explored a minimal set of formal requirements aiming at protecting privacy of both mobile users and consumers. They proposed a privacy-enhanced participatory sensing infrastructure (PEPSI) from a blind extraction technique in identity-based encryption to achieve the anonymity for both mobile users and customers, and utilized a blind matching method to find the sensing reports for a specific task. Nevertheless, Günther et al. [39] showed that PEPSI is vulnerable to collusion attacks across mobile users and customers. As a result, PEPSI cannot protect

the privacy for mobile users. To fix this drawback, they extended the privacy and security model and proposed a generic construction from identity-based encryption. They also presented concrete instantiations from anonymous identity-based encryption schemes.

To protect the location privacy, Christin et al. [40] investigated the location privacy of mobile users and presented a decentralized and collaborative mechanism to protect the path of mobile users, in which the mobile users exchange the sensing data when they physically meet. Zhang et al. [41] raised a concern on location privacy and demonstrated that sensitive contexts are vulnerable to adversaries exploiting spatio-temporal correlations in the behavior of mobile users. Thus, they modeled the potential correlations in a conditional random field, and preserved the location privacy for mobile users by filtering a user's sensing data. Sun et al. [42] introduced SecureFind, a crowdsourced object-finding system offering strong object security to the object owner and strong location privacy to mobile users. SecureFind allows mobile users to generate an indistinguishable dummy tag for the service provider and other mobile detectors. In SecureFind, only the object owner can learn the location of mobile users under a dynamic pseudonym, even the service provider cannot identify the tag or learn any knowledge about the location.

Subsequently, more cryptographic techniques are utilized in MCS for privacy preservation. Wang et al. [43] proposed ARTSense, a framework to achieve trust without identity in MCS, consisting of a privacy-preserving provenance model, an anonymous reputation management scheme and a data trust assessment scheme. This solution does not require trusted third party and enforces both positive and negative reputation updates. Li et al. [44] proposed a blacklist-based anonymous authentication scheme to achieve the anonymous access control for MCS scenarios. Similar to [44], Zhou et al. [45] introduced a generalized efficient batch cryptosystem to achieve both batch encryption and batch decryption from any public key encryption. This cryptosystem is also extended to support fine-grained multi-receiver multi-file sharing, file authority transfer and multiple file owners' settings for addressing secure multi-file sharing in cloud-assisted MCS [46]. Qiu et al. [47] presented a k-anonymous privacy-preserving scheme for mobile sensing that achieves strong privacy preservation for mobile users and high data quality, by integrating a data coding technique and a message transfer strategy.

Liu et al. [48] proposed a large-scale concurrent data anonymous batch verification scheme for mobile healthcare crowdsensing based on an improved certificateless aggregate signature. The proposed scheme can authenticate all sensing bioinformation at once in a privacy-preserving way. The individual data generated by different users can be verified in batch, while the actual identities of participants are hidden. Li et al. [49] studied how to protect bid privacy in a temporally and spatially dynamic MCS system. Following the classical VickreyClarkeGroves auction, a scalable grouping-based privacy-preserving

participant selection scheme is designed where participants are grouped into multiple participant groups and then auctions are organized within groups via secure group bidding. By leveraging Lagrange polynomial interpolation to perturb participants' bids within groups, participants' bid privacy is preserved. Zhang et al. [50] designed two general dataset purchasing frameworks, named CROWDBUY and CROWDBUY++, based on crowdsourcing, with which a buyer can efficiently buy desired data from available mobile users with quality guarantee in a way respecting users' data ownership and privacy. Wu et al. [51] proposed a dynamic trust relationships aware data privacy protection (DTRPP) mechanism for mobile crowdsensing. In this mechanism, combining key distribution with trust management, the trust value of a public key is evaluated according to both the number of supporters and the trust degree of the public key. The trust value is estimated from the accuracy of the public key provided by the encountering nodes. DTRPP achieves the dynamic management of nodes and estimates the trust degree of the public key. Liu et al. [52] designed an enhanced secure certificateless privacy-preserving verifiable data authentication scheme for MCS. The proposed scheme provides unconditional anonymous data authentication service for MCS, by deploying an improved certificateless ring signature as the cryptogram essential, in which the big sensing data should be signed by one of legitimate members in a specific group and could be verified without exposing the actual identity of the participant. Rahaman et al. [53] designed the first provably secure verifier-local revocation-based group signature scheme that supports sublinear revocation, named sublinear revocation with backward unlinkability and exculpability (SRBE). To achieve this performance gain, SRBE introduces time bound pseudonyms for the signer. By introducing low-cost short-lived pseudonyms with sublinear revocation checking, SRBE drastically improves the efficiency of the group signature primitive.

2.2.2 Privacy-enhanced Task Allocation

To find proper mobile users to perform crowdsensing tasks, many task allocation mechanisms have been proposed to allow the service provider to recruit mobile users effectively. The reputation-based approaches [33, 43, 54] are widely used to evaluate the trustworthiness of mobile users. Ren et al. [54] proposed a social aware-crowdsourcing and reputation management scheme to select proper mobile users for task participation, and a report assessment and rewarding scheme to measure the quality of sensing reports and allocate rewards based on the assessed report quality. To achieve better accuracy, Kazemi et al. [55] defined reputation scores to represent the probability that a mobile user can perform a task correctly, and a confidence level to state that a task is acceptable if its confidence is higher than a given threshold. Kazemi and Shahabi [37] focused on spatial task assignment

for spatial crowdsourcing, in which the service provider allocates tasks using greedy, least location entropy priority or nearest neighbor priority algorithms based on the locations of mobile users. However, the locations of mobile users are disclosed to the service provider. To hide their locations, To et al. [56] introduced a framework to protect the locations of mobile users based on differential privacy and geocasting. This framework provides heuristics and optimizations to determine effective geocast regions for reaching high task assignment ratio with low overhead. To keep location privacy in spatial crowdsourcing, Shen et al. [57] proposed a secure task assignment protocol by utilizing additive homomorphic encryption. It focuses on the location privacy of mobile users in a semi-honest adversary model. To et al. [58] introduced a special type of spatial crowdsourcing, in which the mobile users report their locations to the service provider and thereafter the latter assigns each task in proximity to mobile users with the aim of maximizing the overall number of assigned tasks, and exploited the spatial properties of the space to address the maximum task assignment problem in spatial crowdsourcing.

Xiong et al. [59, 60] investigated on the energy-efficient task allocation. Specifically, Xiong et al. [59] proposed an energy-efficient mobile crowdsensing framework to ensure the required number of mobile users returning the reports and the minimized number of redundant tasks allocated. Subsequently, Xiong et al. [60] defined a spatial-temporal coverage metric for MCS and proposed a generic task allocation framework in energy-efficient Piggyback crowdsensing task model, in which the task allocation is optimized with various incentives. In addition, data quality is critical to be guaranteed in MCS. Wang et al. [61] proposed a novel compressive crowdsensing framework by combining compressive sensing, Bayesian inference and active learning techniques for ensuring the quality of data. Liu et al. [62] introduced a concept of quality-of-information (QoI) to evaluate data granularity and presented a QoI-aware energy-efficient scheme to optimize QoI in crowdsensing tasks. Zhang et al. [63] presented a quality-aware sensing framework to achieve considerable sensing coverage in budget-constrained MCS appications. Social-based task allocation is introduced by Amor et al. [64], and a collaborative crowdsourcing approach called SocialCrowd is proposed by leveraging the relationships in social networks. SocialCrowd organizes a crowd of mobile users into teams while preventing data leakage between competing teams. A clustering algorithm is proposed to discover all possible groups of mobile users, and a ranking mechanism is presented based on the semiring approach in the area of soft constraints programming. To preserve the privacy of mobile users and achieve accurate task allocation simultaneously, Kandappu [65] proposed a privacy-preserving crowdsourcing platform and a novel use selection algorithm, which selects the best subset of users for a given survey to meet the balance among cost, accuracy and privacy. Pournajaf et al. [66] identified different task management approaches in MCS, and assessed the threats to

Table 2.1: Comparison on Related Works in Task Allocation

Profiles	Reputation	Quality	Spatial /Temporal	Energy	Location Privacy	User Privacy
[37, 71]	×	×	√	×	×	×
[54, 55]	√	√	×	×	×	×
[56, 57, 58]	×	×	√	×	√	×
[59]	×	×	×	√	×	√
[60]	×	×	√	√	×	×
[61, 63]	×	√	√	×	×	×
[62]	√	√	×	√	×	×
[64]	×	×	×	×	×	√
[65]	×	√	×	√	×	√
[72]	×	√	×	×	×	√

user's privacy, including task tracing attacks, location-based attacks and malicious tasking. To et al. [67] presented a framework for assigning tasks to mobile users in an online manner without corrupting the location privacy of mobile users and customers based on geo-indistinguishability, and then devise techniques to quantify the probability of reachability between a customer and a mobile user, given their perturbed locations. Xiao et al [68] investigated secure mobile crowdsensing and presented how to use deep learning methods, such as stacked autoencoder, deep neural network, and convolutional neural network to improve the MCS security approaches including authentication, privacy protection, faked sensing countermeasures, intrusion detection and anti-jamming transmissions in MCS. Chen et al. [69] investigated the influence of sensing data correlation on differential privacy protection for MCS systems, and explored the perturbation mechanisms from two different perspectives. From a protector's perspective, based on the Bayesian Network, the classical definition of differential privacy is used to deduce the scale parameter; and from an adversary's perspective, the importance of the maximum correlated group is analyzed to compute the Bayesian differential privacy leakage. Sei et al. [70] proposed a new anonymous data-collection scheme to estimate data distribution accurately. Using simulations with synthetic and real datasets, they proved that the proposed method can reduce the mean squared error and the JensenShannon divergence by more than 85% compared with other existing schemes.

2.2.3 Secure Crowdsensed Data Collection

The success of MCS strongly depends on the quality of sensing data generated by mobile users. The quality can be guaranteed if mobile users are gratified for their contributions on tasks and security policies for data protection. For example, in crowdsourced traffic monitoring, the accuracy of traffic estimation relies on the number and quality of sensing reports, however, the more reports submitted by a mobile user, the larger the probability of the mobile user being traced or identified. Varshney et al. [73] modeled the trade-offs among task fulfillment quality, privacy preservation against collusion attacks and the cost of data collection for mobile users. To address these trade-offs, Kajino et al. [74] defined a user-private quality control problem and proposed a user-private latent class protocol from decentralized secure computation, in which a customer can estimate the true results with privacy preservation of mobile users. To preserve users' privacy and improve prediction accuracy, He et al. [75] proposed a privacy-preserving upload mechanism that satisfies diverse privacy requirements of mobile users and guarantees the quality of traffic estimation. This mechanism formalizes the upload decision process under an incomplete information game model, where each mobile user autonomously decides whether to upload or not to balance the trade-off between traffic service quality and location privacy. In crowdsensed map generation, Chen et al. [76] presented a systematic participatory-sensing-based high-quality map generation scheme to meet the privacy demand of individual users. This scheme addresses three major challenges, namely, how to quantify the privacy leakage of mobile users, how to generate theoretically-proven map using the unorganized points, and how to design map generation scheme robust to GPS error. Chang et al. [77] investigated the local data collection and proposed an innovative scheme to accurately estimate the global regression model without knowing local private data even when a large portion of outliers are present. All the information exchanged among mobile users is in an aggregated and privacy-preserving way and only the aggregated reports are submitted to the service provider, which solidly preserves local data confidentiality. Subsequently, Gong et al. [72] mentioned that the aforementioned solutions only address the trade-offs for restricted instances, and proposed a privacy-preserving off-line statistical collection approach to reliably compute the required statistics from a dynamic group of potentially malicious mobile users based on a distributed statistical collection protocol. Hu et al. [78] proposed an integrated strategy to enhance data trustworthiness and defend against the internal threats for mobile crowdsourcing. The strategy integrates several effective methods, including an evaluation scheme for the attribute relevancy and familiarity of participants, a trust relationship establishment method, a group division strategy based on attributes and metagraph, and a core-selecting based incentive mechanism.

2.2.4 Privacy-preserving Data Analysis

How to generate crowdsensing results without leaking any information about individual reports is a challenging problem in MCS. To address this issue, Erlingsson et al. [79] proposed randomized aggregatable privacy-preserving ordinal response (RAPPOR) to achieve crowdsourced statistics for mobile users anonymously with strong privacy protection. RAPPOR is designed to allow the service provider to perform statistics on the sensing data, e.g., categories, frequencies, histograms and other set statistics. PAPPOR also offers a strong security guarantee on the sensing data based on local differential privacy to restrict the exposure of private information. Wang et al. [80] proposed a data aggregation scheme with personalized privacy preservation by utilizing the sparsity of Bloom filter to protect the security levels of mobile users during the aggregation process. The service provider permits to derive unbiased statistical information (histogram) from crowdsensed data, while no adversary can learn the correlation between the privacy levels of mobile users and data values. Chen et al. [81] introduced a group management protocol to guarantee differential privacy of personal data and support user dynamics, data integrity verification and fault tolerance by leveraging a future message buffering mechanism. Wang et al. [82] defined a concept of geo-indistinguishability and proposed a privacy-preserving histogram aggregation mechanism for fine-grained and high-dimensional location-based data. In this mechanism, the occurrence rate of each location is less constrained by the number of locations, and estimation accuracy is much better for big histogram. Chen et al. [83] demonstrated the necessary of protecting mobile users' location privacy and accountability, and presented a participant-density-aware privacy-preserving aggregate statistics scheme. Multi-pseudonym mechanism is utilized to deal with the vulnerability of low participant density, the Paillier cryptosystem and noninteractive zero-knowledge verification are employed to handle Sybil attacks and achieve the accountability of mobile users. Wang et al. [84] investigated the problem of real-time spatio-temporal crowdsrouced data publishing, and designed an online aggregate monitoring scheme over infinite streams with privacy guarantee. Miao et al. [85] proposed a cloud-enabled privacy-preserving truth discovery framework in MCS, which achieves the protection of the sensing reports and the reliability scores derived by the truth discovery approaches. This framework utilizes the homomorphic cryptosystem to realize weighted aggregation on the encrypted reports of mobile users. Chen et al. [86] developed private data aggregation with integrity assurance and fault tolerance for mobile crowdsensing. Specifically, an efficient group management protocol is designed to deal with the participants' dynamic joins and leaves, and a future message buffering mechanism is leveraged to guarantee fault tolerance. The proposed scheme enables continuously obtaining aggregate results and integrity verifications when failures happen.

Table 2.2: Comparison on Related Works in Privacy-Preserving Data Aggregation

References	User Anonymity	Statistics	Differential Privacy	Aggregation
[47]	√	×	×	√
[79]	×	√	√	√
[80]	×	√	×	√
[81]	×	×	√	√
[82]	×	√	×	×
[83]	×	√	×	√
[84]	×	×	√	√
[85]	×	×	×	√

2.2.5 Privacy-aware User Incentive

To prevent privacy leakage for mobile users, privacy-aware incentive mechanisms in MCS have attracted quite a few attentions, some of which leverage auction mechanisms. Zhang et al. [87] proposed a secure and dependable auction scheme by integrating game theory, logical deductions and cryptography, which is proved secure against dishonest participants under the factor that both mobile users and customers may behave dishonestly for their own benefits. Sun and Ma [88] presented a heterogeneous user based privacy-preserving verifiable incentive mechanism for online crowdsourcing with a limited budget by utilizing privacy-preserving verifiable auction schemes. Dimitriou and Krontiris [89] proposed an efficient reverse auction mechanism to offer privacy-preserving user incentive in MCS. Not only could this mechanism guarantee the anonymity of mobile users, but also suggest a reward distribution mechanism from electronic cash and a decentralized scheme that enables mobile users to claim their reward without being linked to their contributed data. Considering data integrity in MCS, Xu et al. [90] presented a universal system model to satisfy the desirable properties for time window dependent tasks in MCS by utilizing a reverse auction mechanism, which models the interactions between the service provider and mobile users. However, these mechanisms fail to preserve bid privacy of mobile users, Jin et al. [91] proposed a differentially private (DP) incentive mechanism to preserve the privacy of each mobile user's bid against curious entities. Based on the single-minded reverse combinatorial auction, this mechanism is differentially private, approximately truthful, individual rational, and computationally efficient, which approximately minimizes the service provider's total payment. For specific application, spectrum sharing, Jin and Zhang [92] presented a novel framework for spectrum database administrator to choose spectrum-sensing participants in a differentially privacy-preserving manner. A reverse auction problem is used to evaluate each participant's true cost on tasks accomplishment, and a new formulation

Table 2.3: Comparison on Related Works in Privacy-Preserving User Incentive

References	Auction	reward	Verification	DP	User Privacy
[63, 88]	√	×	√	×	√
[90, 93]	√	×	×	×	√
[91, 92, 100]	√	×	×	√	√
[94, 95, 89, 96, 97, 98]	×	√	×	×	√

is employed to offer differential location privacy. Wang et al. [93] combined off-line and online incentive mechanisms to propose an incentive mechanism to select mobile users statically and determine winners dynamically after bidding. A privacy protection scheme is presented to preserve the privacy of mobile users, and a two-stage auction algorithm is designed to determine the winners in bidding to overcome the unfairness problem and user encouragement.

Other approaches, such as reward/credit distribution based on crowdsensed data evaluation, have been proposed. Li and Cao [94] emphasized the open problem to address the contradiction of incentive and privacy, and proposed two credit-based privacy-aware incentive schemes for mobile sensing, one depends on a trusted third party and the other does not, from blind signatures, partially blind signatures and extended Merkle tree. To distribute the reward, Niu et al. [95] designed an electronic cent scheme and employed it to propose an electronic cent-based privacy-preserving incentive mechanism for encouraging mobile users to participate tasks. Delgado-Segura et al. [96, 97] analyzed three important issues in MCS: user participation, sensing data quality and user anonymity, and their correlations, and proposed a general framework, PaySense, to incentivize user participation and validate the data quality based on users' reputation using the Bitcoin network. Gisdakis et al. [98] proposed a privacy-preserving incentive scheme to fairly remunerate mobile users to provide various incentives, such as micropayments, and offer high privacy guarantee for mobile users. As a result, this scheme can address security, privacy, accountability and incentive provision issues in MCS, simultaneously. Gisdakis et a. [99] proposed a holistic framework to assess sensing reports and sift malicious contributions, while offering adequate incentives to motivate mobile users for high-quality data submission. Jin et al. [100] integrated user incentive, data aggregation and data perturbation mechanisms to design an incentivizing privacy-preserving data aggregation scheme to offer user selection and incentive. This scheme incorporates the reliability of mobile users to generate highly accurate aggregated results, and provides privacy protection for mobile users and desirable accuracy on final perturbed results.

2.3 Summary

In this chapter, we have briefly reviewed the preliminaries, including bilinear map, number-theoretic problems, BBS+ signature, PS signature, proxy re-encryption, zero-knowledge proof and blockchain. Also, we have given a survey on the existing works about security and privacy in MCS, including privacy preservation for mobile users and customers, privacy-enhanced task allocation, secure crowdsensed data collection, privacy-preserving data analysis, and privacy-aware user incentive. From the comprehensive literature review, we are aware that the security and privacy challenges have not solidly resolved currently. In the following chapters, we will introduce several countermeasures to address the critical challenging issues and reach the research objectives of this thesis.

Chapter 3

Strong Privacy-preserving Task Allocation

3.1 Introduction

The development of wireless communications and mobile devices triggers the emergence of mobile crowdsensing [1], in which user-centric mobile sensing and computing devices, e.g., smartphones, in-vehicle devices and wearable devices, are utilized to sense, collect and process data from the environment. This "Sensing as a Service" [101] elaborates our knowledge of the physical world by opening up a new door for data collection and sharing [4]. Due to the increasing popularity of mobile devices, mobile crowdsensing supports a broad range of sensing applications nowadays, ranging from social recommendation, such as restaurant recommendation, parking space discovery and indoor floor plan reconstrction [102], to environment monitoring, such as air quality measurement, noise level detection and dam water release warning. With human intelligence and user mobility, mobile crowdsensing can significantly improve the trustworthiness of sensing data, extend the scale of sensing applications and reduce the cost on high-quality data collection [103].

While mobile crowdsensing makes data sensing appealing than ever, it also brings new challenges towards mobile users, one of which is privacy leakage, indicating that mobile crowdsensing puts the privacy of mobile users at stake [104, 105, 106]. The sensing data collected from the surrounding areas are necessarily people-centric and related to some aspects of mobile users and their social setting: where they are and where they are going; what places they are frequently visited and what they are seeing; how their health status is and which activity they prefer to do. Photos on social events may expose the

social relations, locations or even political affiliations of mobile users [107]. Furthermore, the more sensing tasks mobile users engaged in and the richer data the users contribute to, the higher probability that their sensitive information may be exposed with. Therefore, preserving the privacy of mobile users is the first-order security concern in mobile crowdsensing. If there is no effective privacy-preserving mechanism to protect the private information for mobile users, it is of difficulty to motivate mobile users to join in mobile crowdsensing services. In addition, the sensing tasks may contain sensitive information about the customers who issue them. Some personal information about the customers, such as identities, locations, references and purchase intentions, can be predicted by curious entities from the releasing tasks. For example, a house agency may know Bob desire to buy a house in a particular area if Bob releases tasks to collect traffic condition and noise level in the neighborhood. To preserve the privacy for both customers and mobile users, several privacy-preserving mobile crowdsensing schemes [32, 33, 35, 47] have been proposed by utilizing anonymization techniques. Nevertheless, anonymity is insufficient for privacy preservation, since the mobile users may be traced from travel routes and social relations. It is possible to uniquely identify 35% of mobile users based on their top-two locations and 85% of them from their top-three locations based on a large set of call data records provided by a US nationwide cell operator [108]. Therefore, it is important to explore strong privacy-preserving mechanisms to prevent privacy leakage for customers and mobile users in mobile crowdsensing.

Once all information about mobile users and customers is perfectly preserved, it is impossible for service providers to accurately recruit mobile users for task performing, while task allocation is a critical component in mobile crowdsensing to ensure the quality of sensing results. Different from traditional sensing networks [109, 110], the produced data cannot be predicted as a priori, and their trustworthiness totally depends on the intelligence and behaviors of mobile users. In general, the higher quality the sensing data have, the more efforts and costs the mobile users should pay. Therefore, the set of mobile users would directly impact the quality of sensing data. How to identify the right groups of mobile users to produce the desired data according to the targets of sensing tasks is a complex problem from the service provider's perspective. Geography-based and reputation-based approaches are popular in mobile crowdsensing to allocate tasks to mobile users, but either has its inherent weaknesses. Firstly, reputation-based task allocation mechanisms [33, 43, 54, 111] need a trusted third party (TTP) to perform heavy reputation management and are vulnerable to reputation-linking attacks, in which the anonymous mobile users can be re-identified from their reputations. Secondly, geography-based task allocation schemes can optimize users selection based on their spatial and temporal correlation [112], but unfortunately it also discloses the content of sensing tasks and the locations of mobile users

to the service provider, while location privacy is one of the primary concerns for mobile users in pervasive environments. In summary, privacy preservation and task allocation become a pair of contradictory objectives in mobile crowdsensing.

To resolve this issue, we propose a Strong Privacy-preserving mObile crOwdseNsing scheme (SPOON) supporting location-based task allocation, decentralized trust management and privacy preservation for both mobile users and customers simultaneously [113, 114]. By leveraging the blind signature [23] and randomizable matrix multiplication, we fully prevent the privacy leakage from all sources for both mobile users and customers, including locations, identities and credit points, without scarifying the normal mobile crowdsensing services of service providers, such as task allocation, data filtering and trust management. The main contributions of this chapter are summarized as three folds:

- ▷ We design a privacy-preserving location matching mechanism based on matrix multiplication to allow service providers to allocate sensing tasks based on the sensing areas of tasks and the geographic locations of mobile users. Specifically, the service provider can determine whether a mobile user is in the sensing area of a task from two randomized matrices generated from the sensing area and the user's location. Thus, the service provider can learn the result of location matching, but has no knowledge about the interested areas of customers and the locations of mobile users.

- ▷ By extending the proxy re-encryption and BBS+ signature, we protect the sensitive information for mobile users and customers to prevent privacy leakage, including their identities, credit points, sensing tasks and sensing reports. Specifically, we allow the registered customers and mobile users to anonymously prove their capacities and trust levels to participate in the crowdsensing services and securely perform the sensing tasks without exposing contents of sensing tasks and sensing reports. Besides, to prevent the mobile users from misbehaving for unfair rewards, a trusted authority enables to detect the greedy mobile users and trace their identities.

- ▷ We introduce a privacy-preserving credit management mechanism for mobile users, in which mobile users are able to prove their trustworthiness without the exposure of credit points and the management of centralized servers. In particular, it supports the positive and negative updates of credit points for mobile users based on the contributions on the tasks. In addition, multiple service providers can cooperatively maintain a unique trust evaluation system, in the way that mobile users are allowed to participate in the mobile crowdsensing services offered by different service providers using unique credit points.

3.2 Problem Statement

We formally define the system model and threat model, and identify our design goals.

3.2.1 System Model

The mobile crowdsensing service provides customers a people-centric way for data collection from surrounding environment. The architecture consists of three kinds of entities: a service provider, customers and mobile users.

Service Providers: Service providers develop cloud services by themselves or rent the cloud resources offered by cloud service providers. They have sufficient storage and computing resources to provide mobile crowdsensing services. The service providers receive sensing tasks from customers and allocate them to mobile users based on their locations. They collect sensing reports from mobile users, select sensing reports based on the credit points of mobile users and generate sensing results for customers. The service provider also distributes credit points to mobile users for incentive.

Customers: The customers can be individuals, corporations or organizations. They need to accomplish data collection tasks, e.g., to study traffic congestion in a city, pollution level of a creek and satisfactory on public transportation, but they do not have sufficient capabilities to perform tasks by themselves. Thereby, they issue their sensing tasks to the service providers to obtain the sensing results.

Mobile Users: Every mobile user has several mobile devices, e.g., mobile phones, tablets, vehicles and smart glasses. These mobile devices, with rich computational, communication and storage resources, are carried by their owners wherever they go and whatever they do. The mobile users make sure the battery on mobile devices have sufficient power to support their normal functions. The mobile users participate in sensing tasks and utilize their portable devices to collect data from their surrounding areas to fulfill sensing tasks, and report sensing data to the service providers for earning credit points.

3.2.2 Threat Model

The service provider is responsible for offering mobile crowdsensing service to customers, but it may strive to increase the income and violate its privacy policy of data protection. For example, Uber, a crowdsourcing-based ride-sharing service provider, made ride-booking data publicly accessible without the permission of customers in January, 2017, for its own

purpose. Therefore, the service provider is not fully trusted, but honest-but-curious. On one hand, the service provider would honestly perform the mobile crowdsensing service; one the other hand, it may learn a spatio-temporal probability distribution for a specific mobile user and other sensitive information about customers and mobile users, e.g., preference, social relation, political affiliation and purchase intention, from the maintained information, including sensing tasks and sensing reports. Moreover, the employees in service provider may capture and exploit the sensitive information about mobile users.

Mobile users are interested in the privacy about the customers and the other mobile users. In particular, they are willing to know the other mobile users participating in the same tasks, and learn more information about customers they are working for to reach the expectations of customers. Further, mobile users may be greedy for the credit points, such that they may anonymously submit more sensing reports than allowed to warn unfair credit points. In addition, the mobile users may maliciously forge, modify the sensing data or deliver ambiguous, biased sensing data to cheat customers for credit points. These forged or biased data can be discovered using redundancy or truth discovery approaches. The locations are extracted from GPS trusted chips in mobile devices or access points, we assume that mobile users cannot modify their location information.

The external attackers, such as eavesdroppers and hackers, also bring serious security threats towards mobile crowdsensing services. It is possible for an attacker to obtain the identities of the nearby mobile users or customers via physical observation, such that the anonymity may be insufficient for privacy preservation for customers and mobile users. The customers are fully trusted since they are the main beneficiaries of mobile crowdsensing service. They will keep the received crowdsensing results confidential to prevent the exposure of sensitive information of mobile users, such as identities, locations and crowdsensing reports.

3.2.3 Design Goals

To enable strong privacy-preserving mobile crowdsensing under the aforementioned system model and against security threats, SPOON should achieve the following design goals:

- ▷ Location-based Task Allocation: The sensing tasks are allocated to the mobile users in the sensing areas defined by the customers, and other mobile users out of the given areas cannot learn any information about the tasks.
- ▷ Location Privacy Preservation: The locations of mobile users and the sensing areas of sensing tasks would not be exposed to others. The mobile users are only aware

whether they are in the sensing area or not.

- ▷ Data Confidentiality: No entity, except the delegated participants, can obtain the content of releasing tasks or sensing reports, such that the privacy of customers and mobile users would not be disclosed to others.
- ▷ Anonymity of Mobile Users and Customers: The customers, mobile users, the service provider or their collusion are unable to link a sensing report to a mobile user or link a sensing task to a customer. It is even impossible for an attacker to identify whether two sensing reports are generated by the same mobile user or two sensing tasks are issued by the same customer.
- ▷ Privacy-Preserving Credit Management: Credit points are used to represent the reputation of mobile users and encourage them to participate in the mobile crowdsensing activities as rewards. The service provider selects the sensing reports based on the credit points of mobile users and awards credit points to mobile users without knowing the exact credit points of mobile users. The balance of credit points is achieved, which means that it is impossible for the mobile users to forge credit points without being detected, such that the total credit points of a mobile user should be equal to the awarded credit points plus the initial points.
- ▷ Greedy User Tracing: The identities of greedy mobile users, who submit more than one sensing report for the same task in a reporting period, are recovered to prevent the mobile user from awarding unfair credit points.

3.3 SPOON

We propose our SPOON, which is composed of five phases, Service Setup, User Registration, Task Allocation, Data Reporting and Credit Assignment, based on the matrix multiplication, the BBS+ signature [23] and the proxy re-encryption [22].

3.3.1 High-Level Description

We first provide a high-level description of SPOON and its information flow, which is shown in Fig. 3.1. The notions frequently used in SPOON are listed in Table 3.1.

Service Setup: A trusted authority (TA) bootstraps the whole mobile crowdsensing service for the service provider by defining the public parameters $(\mathbb{G}, \mathbb{G}_T, p, g, g_0, g_1, g_2, g_3, h, h_0,$

Table 3.1: Frequently Used Notions

$U_{i\{i\in R\}}$	Set of registered mobile users
$U_{i\{i\in \mathcal{L}\}}$	Set of mobile users in sensing area L
ST	A task issued by a customer
$task$	The detailed content of a task ST
$expires$	The expiration time of a task ST
$area$	The sensing region of a task ST
$L_{m\times n}$	A matrix to represent the service area of the service provider
$\widehat{L}_{m\times n}$	A matrix to represent the sensing area of a task ST
$\widetilde{L}_{m\times n}$	A matrix to represent the current and future locations of a user
$\widehat{M}_{m\times n}$	A random invertible matrix
$\widetilde{M}_{m\times n}$	A random invertible matrix
I	The unique identity of a registrant (mobile user or customer)
P_0	The initial credit point of a mobile user
ϵ	The trust level of a sensing report
γ	The maximum of trust level in a task ST
Q	The credit threshold chosen by a mobile user
A, e, s	The anonymous credential of a mobile user or customer
B, f, t	The anonymous credential of a mobile user with credit point P

$h_1, h_2, h_3, h_4, G, H, \mathcal{G}, \mathcal{H}, \mathcal{F}$) and generates its secret key α and the public key T. The service provider also generates the secret key β and the public key S, and defines a matrix $L_{m\times n}$ to denote the geographic region of its crowdsensing service.

User Registration: The TA registers the mobile users and customers, who are willing to participate in the mobile crowdsensing service. It evaluates the registrant to determine the initial credit point P_0 and interacts with the registrant to generate an anonymous credential (A, e, s, B, f, t). (A, e, s) is used to access the mobile crowdsensing service and (B, f, t) is used to credit management for the registrant. To achieve the anonymity, the ownership of (A, e, s) and (B, f, t) is proved by the registrant for identity authentication and credit evaluation using zero-knowledge proofs, respectively. Besides, RK is assigned to the registrant for the decryption of allocated sensing tasks.

Task Allocation: A customer generates a sensing task ST and sends the message $(c_1, c_2, c_3, expires, \widehat{N}_{n\times n}, \gamma, w, \mathcal{PK}_2)$ to the service provider, which consists of the encrypted task (c_1, c_2, c_3), the expiration time $expires$, the randomized sensing area $\widehat{N}_{n\times n}$, the identity proof $\mathcal{PK}_2$ and other information. The latter releases $(num, expires, \gamma)$ to attract mobile users for participation, where num is the identifier of ST. A mobile user $U_{i\{i\in R\}}$ sends its location $\widetilde{N}_{n\times n}$ and identity proof $\mathcal{PK}_3$ to the service provider. Then, the service provider finds the set of mobile users $U_{i\{i\in\mathcal{L}\}}$ in the sensing area of ST based on two matrices $(\widehat{N}_{n\times n}, \widetilde{N}_{n\times n})$. Since $(\widehat{N}_{n\times n}, \widetilde{N}_{n\times n})$ are randomized matrices, the service provider can learn whether U_i is in the sensing area of ST based on matrix multiplication, but has no information about ST's sensing area and U_i's location. The service provider re-encrypts the ciphertext (c_1, c_2, c_3) to be decryptable for $U_{i\{i\in\mathcal{L}\}}$ using β. Finally, the service provider sends $(num, c_2, c_3, c_4, expires, \gamma, w)$ to $U_{i\{i\in\mathcal{L}\}}$.

Data Reporting: $U_{i\{i\in\mathcal{L}\}}$ encrypts the collected data m_i to generate (D_i, D_i'), and sends the sensing report $(num, D_i, D_i', C_i', X_i, Y_i, Z_i, Q_i, \tau_j, \mathcal{SPK})$ to the service provider, in which C_i' is the commitment on the identity I_i and credit point P_i, X_i is the identifier of this report, Y_i is the identifier of U_i, Z_i is a tag to identify the double-reporting user, Q_i is the claimed credit threshold to show that the number of credit points U_i has is larger than Q_i, τ_j is the current slot for reporting, and $\mathcal{SPK}$ is used to prove the ownership of its credit points P_i. The service provider selects $w-$sensing reports based on the claimed thresholds and forwards the selected reports to the customer. The TA can recover the identity of anonymous mobile user who double-reports sensing reports with the service provider using the double-reporting tag Z_i.

Credit Assignment: The customer evaluates the trustworthiness of each report and returns the corresponding trust level $\epsilon_i \in [-\gamma, \gamma]$ to the service provider. The latter computes the number of credit points awarded to U_i, θ_i, and forwards $(B_i, t_i'', f_i, \theta_i, Y_i)$ to U_i, where

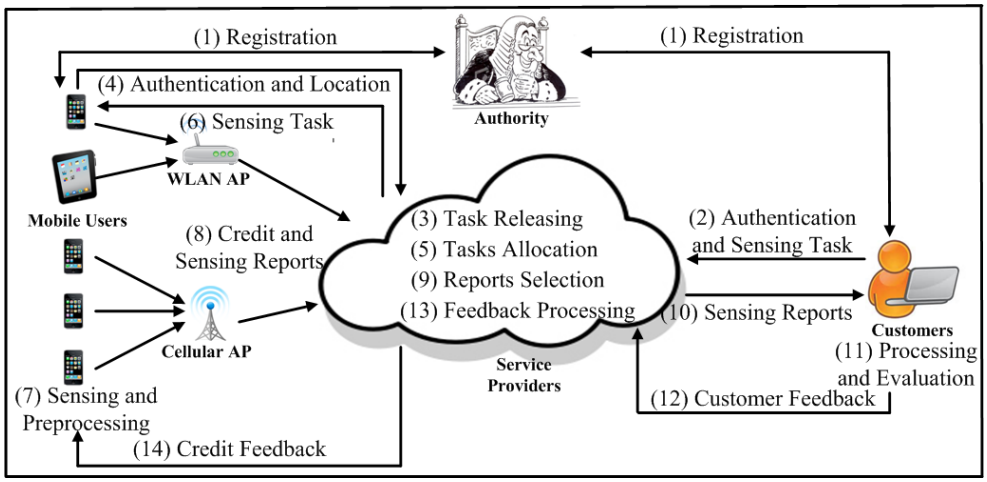


Figure 3.1: Information Flow of Mobile Crowdsensing.

(B_i, t_i'', f_i) is the ticket for awarded credit points θ_i, and Y_i is used to identify the mobile user U_i. Once receiving $(B_i, t_i'', f_i, \theta_i, Y_i)$, U_i updates its credit points $P_i' = P_i + \theta_i$ and the anonymous credential (B_i, f_i, t_i) for the new P_i'.

3.3.2 The Detailed SPOON

We then show the detailed SPOON as follows.

3.3.2.1 Service Setup

Let $(\mathbb{G}, \mathbb{G}_T)$ be two cyclic groups with a prime order p, where p is λ bits, and $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. The authority randomly picks generators $g, g_0, g_1, g_2, g_3, h, h_0, h_1, h_2, h_3, h_4 \in \mathbb{G}$ and computes $G = \hat{e}(g, g)$ and $H = \hat{e}(h, h)$ respectively. The TA also chooses a random value $\mathcal{G} \in \mathbb{G}_T$ and defines a cryptographic hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p$ and a pseudo-random function $\mathcal{F} : \mathbb{Z}_p \times \{0,1\}^* \to \mathbb{Z}_p$. The public parameters *param* are $(\mathbb{G}, \mathbb{G}_T, p, g, g_0, g_1, g_2, g_3, h, h_0, h_1, h_2, h_3, h_4, G, H, \mathcal{G}, \mathcal{H}, \mathcal{F})$. The TA randomly chooses $\alpha \in \mathbb{Z}_p$ as its secret key and calculates the public key $T = g^\alpha$.

To setup the mobile crowdsensing service, the service provider randomly chooses its secret key $\beta \in \mathbb{Z}_p$ and computes $S = h^\beta$ as its public key. It also employs a matrix $L_{m \times n}$

to denote the geographical region that the crowdsensing service can cover according to the longitude and latitude. Each entry in the matrix denotes a small grid in the sensing region, as shown in Fig. 3.2. Assume the longitude of Ontario is from 74.40°W to 95.15°W, the latitude is from 41.66°N to 57.00°N, we can use a 208×154 matrix or 2075×1534 matrix more precisely to represent the Ontario region.

3.3.2.2 User Registration

Either customer or mobile user is required to register at the TA to obtain an anonymous credential, which is used to participate in the crowdsensing service. Each registrant is assigned a unique identity I in the system, which can be the telephone number or mailing address in practise. The registrant picks three random values $s', a, t' \in \mathbb{Z}_p$ to compute $C = g_1^{s'} g_2^a$, $C' = h_1^{t'} h_2^a$, $\widehat{A} = g^a$, and sends $(I, C, C', \widehat{A})$ to the TA, along with the following zero-knowledge proof:

$$\mathcal{PK}_1\{(s', t', a) : C = g_1^{s'} g_2^a \wedge C' = h_1^{t'} h_2^a \wedge \widehat{A} = g^a\}.$$

The TA firstly checks the proof $\mathcal{PK}_1$ for ensuring that $(C, C', \widehat{A})$ are generated correctly. Then, it evaluates the registrant's initial credit point according to its credit record, which is assumed to be P_0. After that, the TA randomly picks $s'', e, t'', f \in \mathbb{Z}_p$ to calculate $A = (g_0 C g_1^{s''} g_3^I)^{\frac{1}{\alpha+e}}$, $B = (h_0 C' h_1^{t''} h_3^I h_4^{P_0})^{\frac{1}{\alpha+f}}$, $RK = \widehat{A}^{\frac{1}{\alpha}}$, and returns $(A, B, s'', t'', e, f, P_0, RK)$ to the registrant through secure channels. Finally, the TA stores the tuple $(I, P_0, \widehat{A})$ in its database.

The registrant computes $s = s' + s'', t = t' + t''$ and checks

$$\hat{e}(A, Tg^e) \stackrel{?}{=} \hat{e}(g_0 g_1^s g_2^a g_3^I, g), \qquad \hat{e}(B, Th^f) \stackrel{?}{=} \hat{e}(h_0 h_1^t h_2^a h_3^I h_4^{P_0}, h).$$

The registrant stores $(A, e, s, B, f, t, a, I, P_0, \widehat{A}, RK)$ secretly on the read-only memory of mobile device.

3.3.2.3 Task Allocation

A customer with registered information $(A, e, s, B, f, t, a, I, P_0, \widehat{A}, RK)$ generates a sensing task to be allocated to mobile users and requests the sensing data slot by slot, where each slot ranges from minutes to days depending on the specific requirements of the sensing task. The statement of the task is defined as $ST = (task, expires, area, \gamma, w)$, which indicate the

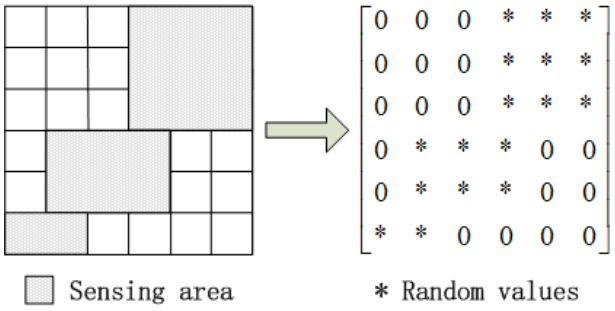


Figure 3.2: Sensing Area and the Matrix $\widehat{L}_{6\times 6}$.

content (what to sense), the expiration time (when to sense), the sensing area (where to sense), the maximum trust level and the number of required reports, respectively. Other attributes (e.g., sensing intervals, acceptance conditions, benefits, reporting periods) can be illustrated in *task*. To protect the content of the task, the customer randomly picks $k, r_1, r_2, r_3 \in \mathbb{Z}_p$ to calculate $u = g^k, c_1 = S^{r_2}$, $c_2 = T^{r_1}$ and $c_3 = (task||u)G^{r_1}H^{r_2}$. Then, the customer generates a matrix $\widehat{L}_{m\times n}$ to indicate the target sensing region *area*. As depicted in Fig. 3.2, for each position in the sensing area, the corresponding entry in $\widehat{L}_{m\times n}$ is set to be a random value chosen from $\mathbb{Z}_p^*$, and the value for a location outside is set to be zero. To mask the sensing area in $\widehat{L}_{m\times n}$, the customer picks $m \times n$ random numbers from $\mathbb{Z}_p^*$ to generate an invertible matrix $\widehat{M}_{m\times n}$ and computes $\widehat{N}_{n\times n} = \widehat{L}^T_{m\times n} \cdot \widehat{M}_{m\times n}$, where $\widehat{L}^T_{m\times n}$ is the transpose of the matrix $\widehat{L}_{m\times n}$. Note that all non-zero entries in $\widehat{L}_{m\times n}$ should be distinct, unless an attacker still can learn the sensing region from $\widehat{N}_{n\times n}$. Finally, the customer keeps k in private and sends $(c_1, c_2, c_3, expires, \widehat{N}_{n\times n}, \gamma, w)$ to the service provider, along with the following zero-knowledge proof:

$$\mathcal{PK}_2\{(A, e, s, a, I) : \hat{e}(A, Tg^e)=\hat{e}(g_0 g_1^s g_2^a g_3^I, g)\}.$$

The service provider checks the validity of the proof $\mathcal{PK}_2$. If yes, it assigns a task identifier *num*, releases $(num, expires, \gamma)$ and stores $(num, c_1, c_2, c_3, expires, \widehat{N}_{n\times n}, \gamma, w)$ in its database.

When a mobile user $U_{i\{i\in R\}}$ with $(A_i, e_i, s_i, B_i, f_i, t_i, a_i, I_i, P_i, \widehat{A}_i, RK_i)$ is willing to participate in crowdsensing activities, it firstly picks a random value $\nu \in \mathbb{Z}_p$ to calculate $\mu = h^\nu$. Then, U_i generates a matrix $\widetilde{L}_{m\times n}$ according to its current location and the places it will visit. For each location U_i will reach, the corresponding entry in $\widetilde{L}_{m\times n}$ is set to be a random value chosen from $\mathbb{Z}_p^*$, and the rest entries are set to be zero. The non-zero entries in $\widetilde{L}_{m\times n}$ should be different. To protect these location information, it also generates a

random invertible matrix $\widetilde{M}_{m\times n}$ by picking $m \times n$ random values from $\mathbb{Z}_p^*$, and calculates $\widetilde{N}_{n\times n} = \widetilde{M}_{m\times n}^T \cdot \widetilde{L}_{m\times n}$. Finally, U_i keeps ν secretly and sends $(\mu, \widetilde{N}_{n\times n})$ to the service provider, along with the following zero-knowledge proof:

$$\mathcal{PK}_3\{(A_i, e_i, s_i, a_i, I_i) : \hat{e}(A_i, Tg^{e_i}) = \hat{e}(g_0 g_1^{s_i} g_2^{a_i} g_3^{I_i}, g)\}.$$

The service provider returns failure if $\mathcal{PK}_3$ is invalid. Otherwise, for each unexpired task, it uses $\widehat{N}_{n\times n}$ to calculate $N_{n\times n} = \widetilde{N}_{n\times n} \cdot \widehat{N}_{n\times n}$ and checks whether $N_{n\times n}$ is zero matrix or not. If $N_{n\times n}$ is non-zero matrix, which means that U_i can match ST, the service provider calculates $c_4 = \hat{e}(\mu, c_1)^{\frac{1}{\beta}}$ and releases $(num, c_2, c_3, c_4, expires, \gamma, w)$ for U_i. If there is no task to match U_i, the service provider responds failure.

When U_i obtains $(num, c_2, c_3, c_4, expires, \gamma, w)$, it decrypts (c_2, c_3, c_4) by using (ν, a_i) as $task||u = c_3 c_4^{-\frac{1}{\nu}} \hat{e}(c_2, RK_i)^{-\frac{1}{a_i}}$. Then, U_i evaluates the task and determines to participate in or abandon this task according to benefit and cost. If U_i accepts the task ST, it starts to perform the sensing work according to the details in $task$. The correctness of $task||u$ is elaborated as follows:

$$\begin{aligned} & c_3 c_4^{-\frac{1}{\nu}} \hat{e}(c_2, RK_i)^{-\frac{1}{a_i}} \\ = \; & c_3 \hat{e}(\mu, c_1)^{-\frac{1}{\beta\nu}} \hat{e}(c_2, RK_i)^{-\frac{1}{a_i}} \\ = \; & (task||u) G^{r_1} H^{r_2} \hat{e}(h^\nu, S^{r_2})^{-\frac{1}{\beta\nu}} \hat{e}(T^{r_1}, g^{\frac{a_i}{\alpha}})^{-\frac{1}{a_i}} \\ = \; & (task||u) G^{r_1} H^{r_2} H^{-r_2} G^{-r_1} \\ = \; & task||u. \end{aligned} \tag{3.1}$$

3.3.2.4 Data Reporting

U_i collects and, pre-processes the data $m_i \in \mathbb{G}_T$ and submits a sensing report to the customer periodically, which includes the collection time, the sensing location and the detailed content. The reporting periods are defined by the customer, and we assume the current slot is τ_j. To prevent attackers from learning m_i, U_i uses u to encrypt m_i as $D_i = u^{\hat{r}_i}$, $D_i' = m_i G^{\hat{r}_i}$, where $\hat{r}_i$ is a value randomly chosen from $\mathbb{Z}_p$. Then, U_i randomly picks $t_i' \in \mathbb{Z}_p$ to compute $C_i' = h_1^{t_i'} h_2^{a_i} h_3^{I_i} h_4^{P_i}$. Next, U_i computes $X_i = \mathcal{H}(num||m_i||\tau_j)$, $v_i = \mathcal{F}_{a_i}(num||I||\tau_j)$, $Y_i = H^{v_i}$ and $Z_i = \hat{e}(g, \widehat{A}_i)\mathcal{G}^{X_i v_i}$. Finally, U_i chooses a credit threshold Q_i and sends the report $(num, D_i, D_i', C_i', X_i, Y_i, Z_i, Q_i, \tau_j)$ to the service provider,

along with the following zero-knowledge proof:

$$\mathcal{SPK}\left\{\begin{array}{c}(B_i, f_i, t_i, t_i', a_i, I_i, P_i, v_i):\\ \hat{e}(B_i, Th^{f_i})=\hat{e}(h_0 h_1^{t_i} h_2^{a_i} h_3^{I_i} h_4^{P_i}, h)\wedge\\ C_i' = h_1^{t_i'} h_2^{a_i} h_3^{I_i} h_4^{P_i}\wedge\\ P_i > Q_i\wedge\\ Y_i = H^{v_i}\wedge\\ Z_i = \hat{e}(g, \widehat{A}_i)\mathcal{G}^{X_i v_i}\end{array}\right\}(num).$$

The service provider returns failure if $\mathcal{SPK}$ is invalid; otherwise, the service provider checks whether there is another report $(num, \widetilde{D}_i, \widetilde{D}_i', \widetilde{C}_i', \widetilde{X}_i, Y_i, \widetilde{Z}_i, \widetilde{Q}_i)$ that has the same Y_i and different $\widetilde{X}_i$ with the new received report $(num, D_i, D_i', C_i', X_i, Y_i, Z_i, Q_i)$. If yes, the service provider computes and sends $W = (\frac{\widetilde{Z}_i^{X_i}}{Z_i^{\widetilde{X}_i}})^{\frac{1}{X_i - \widetilde{X}_i}}$ to the TA, and the TA can find the mobile user's identity I_i by utilizing $\widehat{A}_i$ in the database to check $W = \hat{e}(g, \widehat{A}_i)$. In other words, the identity of the greedy mobile user is recovered by the TA if it submits two different sensing reports in a reporting slot. Then, according to the claimed thresholds, the service provider chooses w reports that have top-w thresholds, and releases them for the customer. Note that the mobile users, whose reports are not selected, can increase their thresholds in the next reporting slot $\tau_j + 1$.

When the customer retrieves the reports, it can decrypt them using the stored k as $m_i = D_i' \hat{e}(g, D_i)^{\frac{1}{k}}$ one by one.

3.3.2.5 Credit Assignment

After the customer obtains the sensing result, it evaluates the trustworthiness of each report and responds the corresponding trust level to the service provider. The trust level of m_i is defined as $\epsilon_i \in [-\gamma, \gamma]$. If ϵ_i is positive, m_i is trustworthy, otherwise, m_i is incredible.

Upon receiving trust levels, the service provider randomly picks $t_i'', f_i \in \mathbb{Z}_p$ to compute $\theta_i = INT(\epsilon_i Q_i)$, $B_i = (h_0 h_1^{t_i''} C_i' h_4^{\theta_i})^{\frac{1}{\beta + f_i}}$, and releases $(B_i, t_i'', f_i, \theta_i, Y_i)$ for U_i, where $INT(x)$ is the nearest integer function.

U_i retrieves $(B_i, t_i'', f_i, \theta_i, Y_i)$ from the service provider, computes $t_i = t_i' + t_i''$, $P_i' = P_i + \theta_i$ and checks whether $\hat{e}(B_i, Sh^{f_i})=\hat{e}(h_0 h_1^{t_i} h_2^{a_i} h_3^{I_i} h_4^{P_i'}, h)$ or not. If yes, U_i uses the new tuple (B_i, f_i, t_i, P_i') to replace the previous one and stores them with $(A_i, e_i, s_i, a_i, T_i, \widehat{A}_i, RK_i)$. Meanwhile, U_i updates P_i' in the read-only memory, which can be used to show the credit

points in the future crowdsensing activities. Further, since (B_i, f_i, t_i, P_i') are managed by U_i, U_i enables to prove the ownership of (B_i, f_i, t_i) cross service providers. The credit points awarded by different service providers can be accumulated and U_i can prove the credit points to multiple service providers during the participation of mobile crowdsensing services offered by different service providers.

3.4 Security Analysis

We show that SPOON satisfies five security goals defined in 3.2.3: location privacy, anonymity, data confidentiality, credit balance and greedy user tracing.

3.4.1 Location Privacy

The sensing region of a task is represented as a matrix $\widehat{L}_{m\times n}$, which is randomized by a random matrix $\widehat{M}_{m\times n}$ to generate $\widehat{N}_{n\times n}$. The location of the mobile user is transformed to be $\widetilde{N}_{n\times n}$. Having two matrices $\widehat{N}_{n\times n}$ and $\widetilde{N}_{n\times n}$, the service provider cannot learn any information about the location of the mobile user and the sensing area of the task. The service provider computes $N_{n\times n} = \widehat{N}_{n\times n} \cdot \widetilde{N}_{n\times n}$. If there is no overlapping between the sensing area of task and the location of mobile user, $N_{n\times n}$ must be zero matrix. If one overlapping grid exists, whose corresponding entry is $\widehat{L}_{ij}$ in $\widehat{L}_{m\times n}$ and is $\widetilde{L}_{ij}$ in $\widetilde{L}_{m\times n}$, respectively, the entries in j-row of $\widehat{N}_{n\times n}$ are nonzero, as well as the entries in j-column of $\widetilde{N}_{n\times n}$. Thus, the service provider enables to know that there are some overlapping locations on the j-column of the sensing area, while it is unable to distinguish which location is overlapped from m locations. Further, $\widehat{N}_{n\times n} \cdot \widetilde{N}_{n\times n}$ and $\widetilde{N}_{n\times n} \cdot \widehat{N}_{n\times n}$ cannot give more information to the service provider. The results are the same if the overlapping grids are more than one. Therefore, the sensing area and the location of mobile user would not be exposed to the service provider and other entities.

3.4.2 Data Confidentiality

We aim to ensure that only the mobile users whose locations can match the sensing area have the capacity to recover the corresponding sensing task. In SPOON, the adversaries may be the service provider, unmatched mobile users and external attackers. To resist these adversaries, the task protection consists of two stages. In the first stage, the sensing task is encrypted by the customer under the public keys of the TA and the service provider;

in the second one, the service provider partially decrypts the ciphertext using its secret key and then re-encrypts the result for the matched mobile users. Therefore, we demonstrate the task confidentiality in the following two procedures:

- ▷ Firstly, the first-stage ciphertext should not be entirely decryptable for the service provider or the mobile users. To be specific, given the first-stage ciphertext (c_1^*, c_2^*, c_3^*) and two plaintexts $(task_1||u_1, task_2||u_2)$, if an adversary can distinguish which one out of $(task_1||u_1, task_2||u_2)$ is the plaintext of (c_1^*, c_2^*, c_3^*), we show how to construct a simulator $\mathcal{S}$ to solve the $q-$DBDHI problem [22].

  Given the simplified $q-$DBDHI tuple $g, T_1 = g^{z_1}, T_2 = g^{z_2} \in \mathbb{G}, Q \in \mathbb{G}_T$, the simulator $\mathcal{S}$'s goal is to determine whether $Q = \hat{e}(g,g)^{\frac{z_1}{z_2}}$ via interactions with the adversary. $\mathcal{S}$ sets $T = T_1$. The adversary possessing the secret key of the service provider, β, can query any chosen message $task||u$ to the simulator $\mathcal{S}$ to obtain the corresponding the ciphertext. Then, $\mathcal{S}$ picks two messages $(task_1||u_1, task_2||u_2)$ and a random bit $b \in \{0,1\}$ to compute the challenge $(c_1^*, c_2^*, c_3^*) = (S^{r_2}, T_2, (task_b||u_b)QH^{r_2})$, where r_2 is a random value chosen from $\mathbb{Z}_p$, and returns $(task_1||u_1, task_2||u_2)$ to the adversary, along with (c_1^*, c_2^*, c_3^*). Finally, the adversary returns $\hat{b} \in \{0,1\}$ to $\mathcal{S}$. If $\hat{b} = b$, $\mathcal{S}$ can address the simplified $q-$DBDHI problem as $Q \stackrel{?}{=} \frac{c_3^*}{(task_b||u_b)\hat{e}(c_1^*,h)^{-\frac{1}{\beta}}}$.

  The task confidentiality against the adversary, who possesses α, also relies on the simplified $q-$DBDHI problem, given $h, T_1 = h^{z_1}, T_2 = h^{z_2} \in \mathbb{G}, Q \in \mathbb{G}_T$, The proof is the same as that above with one difference that the challenge is $(c_1^* = T_2, c_2^* = T_1^{r_1}, c_3^* = (task_b||u_b)QG^{r_1})$, where r_1 is a random value chosen from $\mathbb{Z}_p$. Finally, $\mathcal{S}$ can address the simplified $q-$DBDHI problem as $Q \stackrel{?}{=} \frac{c_3^*}{(task_b||u_b)\hat{e}(c_2^*,g)^{-\frac{1}{\alpha}}}$.

- ▷ Secondly, the sensing task should only be recovered by the matched mobile users from the second-stage ciphertext. To prevent unmatched mobile users from learning the content of sensing task, the service provider encrypts the sensing task with the temporary public key μ using the proxy re-encryption scheme [22]. Therefore, the security of the second-stage ciphertext can be reduced to the $q-$DBDHI assumption as well.

To guarantee the confidentiality of sensing reports, each mobile user employs the proxy re-encryption scheme [22] to encrypt m_i under the temporary public key $u = g^k$, which is distributed to the mobile users along with the sensing task. The decryption key k is kept by the customer secretly. Therefore, the confidentiality of m_i directly depends on the semantic security of proxy re-encryption scheme, which can be reduced to the simplified $q-$DBDHI assumption [22].

3.4.3 Anonymity

The anonymity of the mobile user is defined via the game in which the adversary cannot distinguish an honest mobile user out of two under the extreme condition that all other interactions are specified by the adversary. We prove that the mobile user's identity is preserved properly, provided the DDH assumption [115] holds. Specifically, if there exists an adversary $\mathcal{A}$ that can identify an honest mobile user out of two challenging identities, we show how to construct a simulator $\mathcal{S}$ to solve an instance of the DDH problem. That is, given a tuple $T_1, T_2, T_3, T_4 \in \mathbb{G}_T$, $\mathcal{S}$ can tell whether exists (z_1, z_2), such that $T_2 = T_1^{z_1}$, $T_3 = T_1^{z_2}$, $T_4 = T_1^{z_1 z_2}$. $\mathcal{S}$ generates $(param, S, T)$, picks two identities (I_0, g^{a_0}), (I_1, g^{a_1}), where $a_0, a_1 \in \mathbb{Z}_p$, and sends them to $\mathcal{A}$. $\mathcal{S}$ acts on behalf of the users I_0 and I_1 to register at the TA. $\mathcal{S}$ then interacts with $\mathcal{A}$ in the following interactions:

- ▷ $\mathcal{S}$ acts as I_0 honestly to submit the location information. For I_1, in the j-th query, $\mathcal{S}$ randomly chooses $\mu_j \in \mathbb{G}$ and simulates the zero-knowledge proof $\mathcal{PK}_3$ to prove its identity interacting with $\mathcal{A}$.
- ▷ $\mathcal{S}$ honestly acts on behalf of I_0 to report the data. For I_1, $\mathcal{S}$ sets $H = T_1$, $\mathcal{G} = T_2$. For the j-th query, $\mathcal{S}$ randomly chooses X_j, $v_j \in \mathbb{Z}_p$ and computes $Y_j = T_1^{v_j}$, $Z_j = \hat{e}(g, g^{a_1}) T_2^{X_j v_j}$. $\mathcal{S}$ simulates the zero-knowledge proof $\mathcal{SPK}$ and sends $(X_j, Y_j, Z_j, \mathcal{SPK})$ to $\mathcal{A}$, along with a random sensing report.

$\mathcal{S}$ picks a random bit $b \in \{0, 1\}$. If $b = 0$, $\mathcal{S}$ honestly reports the data acts as I_0. If $b = 1$ and $\mathcal{S}$ randomly chooses $X_1 \in \mathbb{Z}_p$ and calculates $\mathcal{G} = T_2$, $Y_1 = T_3$, $Z_1 = \hat{e}(g, g^{a_1}) T_4^{X_1}$. Then, $\mathcal{S}$ simulates $\mathcal{SPK}$ and a sensing report, and sends them to $\mathcal{A}$. It is easy to see that the simulation is perfect if $log_{T_1} T_4 = log_{T_1} T_2 log_{T_1} T_3$; otherwise, it contains no information about I_0 and I_1.

Finally, $\mathcal{A}$ returns $\hat{b}$. If $\hat{b} = b$, $\mathcal{S}$ can confirm that there exists (z_1, z_2), such that $T_2 = T_1^{z_1}$, $T_3 = T_1^{z_2}$, $T_4 = T_1^{z_1 z_2}$. Thus, $\mathcal{S}$ resolves the DDH problem.

In the proof of customer's anonymity, a simulator $\mathcal{S}$ simulates the transcript of the zero-knowledge proof of the signature (A, e, s), $\mathcal{PK}_2$, to interact with the adversary $\mathcal{A}$. Since $\mathcal{S}$ can perfectly simulates $\mathcal{PK}_2$, the adversary cannot obtain any identity information about the customer, such that it is impossible to distinguish an honest customer from two for $\mathcal{A}$. Therefore, the customer's anonymity can be fully guaranteed.

3.4.4 Credit Balance

Credit balance means that no one can own the credit points more than the initial credit points plus the credit points awarded by service providers. This is the most significant requirement for credit management from the respective of security. Assume P_0 be the initial credit points and θ_j be the earned points from the service provider in the j-th query. If the adversary $\mathcal{A}$ at most makes $\hat{R}$ reporting queries, and owns final credit points P_f, where $P_f > P_0 + \sum_{j=1}^{\hat{R}} \theta_j$, while service providers do not identify the double-reporting, there must exist a simulator $\mathcal{S}$ to conduct a forgery attack on the underlying BBS+ signature [23].

Firstly, we assume that the zero-knowledge proofs $\mathcal{PK}_1$, $\mathcal{PK}_2$, $\mathcal{PK}_3$ and $\mathcal{SPK}$ are sound. That is, there exist extract algorithms $\mathcal{EX}_1$, $\mathcal{EX}_2$, $\mathcal{EX}_3$ and $\mathcal{EX}_S$ to obtain the witnesses of the zero-knowledge proofs, respectively.

Then, we show the simulator $\mathcal{S}$ that interacts with $\mathcal{A}$. $\mathcal{S}$ generates the public parameters $param$, the public keys (T, S) and the secret keys (α, β), and is allowed to access the signature oracle $\mathcal{SO}$ to get the BBS+ signature of an input. $\mathcal{S}$ sends $(param, S, T)$ to $\mathcal{A}$ and interacts with $\mathcal{A}$ as follows:

- ▷ $\mathcal{A}$ randomly chooses $C, C', \widehat{A} \in \mathbb{G}$, and generates the proof $\mathcal{PK}_1$ and sends them to $\mathcal{S}$. $\mathcal{S}$ extracts the witness (s', t', a) from $\mathcal{PK}_1$ using $\mathcal{EX}_1$, and then picks a random credit point P_0 and queries the signature oracle $\mathcal{SO}$ to obtain (A, e, s) and (B, f, t). Finally, $\mathcal{S}$ calculates $s'' = s - s'$, $t'' = t - t'$ and $RK = \widehat{A}^{\frac{1}{\alpha}}$, and returns $(A, e, s'', B, f, t'', P_0, RK)$ to $\mathcal{A}$.

- ▷ For the j-th query, $\mathcal{A}$ picks a random $C'_j \in \mathbb{G}$ and executes $\mathcal{SPK}$ with $\mathcal{S}$. $\mathcal{S}$ utilizes $\mathcal{EX}_S$ to extract the witness $(B_j, f_j, t_j, t'_j, a_j, I_j, P_j, v_j)$. If (B_j, f_j, t_j) is not an output of $\mathcal{SO}$, it is a forgery of the BBS+ signature. Otherwise, $\mathcal{S}$ queries $\mathcal{SO}$ to obtain a signature (B_j, f_j, t_j) on input $(a_j, P_j + \theta_j, I_j)$. $\mathcal{S}$ receives (B_j, f_j, t_j) and computes $t''_j = t_j - t'_j$, and returns (B_j, f_j, t''_j) to $\mathcal{A}$.

Finally, assume $\mathcal{A}$ executes $\hat{R}$ queries. $\mathcal{A}$ wins the game if it can prove $P_f > P_0 + \sum_{j=1}^{\hat{R}} \theta_j$. However, if $P_f > P_0 + \sum_{j=1}^{\hat{R}} \theta_j$, $\mathcal{A}$ must have conducted a forged BBS+ signautre or double-reported the data. While the BBS+ signature is secure under the $q-$SDH assumption [23], $\mathcal{A}$ cannot forge a BBS+ signature, unless the $q-$SDH assumption [23] does not hold. If $\mathcal{A}$ double-reports the sensing data, it must generate another $\widetilde{Z}_i$, which is unequal to the previous Z_i, in the same time slot. Due to the soundness of zero-knowledge

Table 3.2: Computational Overhead of SPOON

Phase	User Registration		Task Allocation		
	Authority	User	Customer	Provider	User
Point Multiplication	16	19	11	12	9
Point Addition	12	13	5	8	5
Bilinear Map	0	4	1	1	2
Exponentiation in $\mathbb{G}_T$	0	0	6	15	8
Running Time (ms)	83.429	293.372	100.123	138.529	154.980

Phase	Data Reporting			Credit Assignment	
	Customer	Provider	User	Provider	User
Point Multiplication	0	19	25	3	5
Point Addition	0	14	16	3	5
Bilinear Map	1	5	2	0	2
Exponentiation in $\mathbb{G}_T$	1	19	15	0	0
Running Time (ms)	56.415	197.324	203.129	15.643	130.448

proof protocol, $Z_i = \hat{e}(g, \widehat{A}_i)\mathcal{G}^{X_i v_i}$ is the only valid Z_i to accompany the specific report identified by X_i and Y_i. Since X_i should be different for two reports, $\hat{e}(g, \widehat{A}_i)$ would be obtained as long as the proof is valid. We assume the proof $\mathcal{SPK}$ is sound. Thus, the success probability of double-reporting for $\mathcal{A}$ is negligible. Therefore, the probability to obtain $P_f > P_0 + \sum_{j=1}^{\hat{R}} \theta_j$ is negligible if the $q-$SDH assumption holds.

3.4.5 Greedy User Tracing

Greedy user tracing consists of two objectives, namely, slandering and hiding. Slandering means that an attacker cannot slander an honest mobile user, and hiding means that a greedy user must be identified by the TA. For the slandering, the attacker releases pieces of reporting transcripts that can link to other reports submitted by an honest mobile user. It is infeasible for the attacker to compute the tracing information about an honest mobile user since the proof $\mathcal{SPK}$ is sound. Therefore, no attacker enables to slander an honest mobile user. In terms of the hiding, the attacker is required to generate different pieces of tracing information without being traced. However, it is impossible for a greedy mobile user to compute Z_i if the pseudo-random function $\mathcal{F}$ is correct.

3.5 Performance Evaluation

Here, we evaluate the performance of our SPOON in terms of computational and communication overheads, and analyze privacy rate and accuracy rate for credit management.

3.5.1 Computational Overhead

We demonstrate the computational overhead of our SPOON by counting the number of the time-consuming cryptographic operations, such as point multiplication, point addition, bilinear map and exponentiation in $\mathbb{G}_T$. Here we only show four kinds of operations because other operations, e.g., multiplication in $\mathbb{G}_T$, addition, multiplication and inverse operations in $\mathbb{Z}_p$, are not comparable with these four operations. Besides, since the bilinear map is the most time-consuming operation in cryptographic calculations, we utilize the pre-processing technique to reduce the computational burden for each entity. Specifically, the TA pre-computes the bilinear maps $\{E_i\}_{i=0}^4, \{F_i\}_{i=0}^4, K, K_0', \{K_i\}_{i=0}^3$ in service setup phase as shown in Appendix A, and the bilinear maps $\{\hat{e}(g, \widehat{A}_i)\}_{i=0}^N$ in user registration phase, where N is the number of registrants. The mobile user U_i also can pre-compute $\hat{e}(g, \widehat{A}_i)$ in user registration phase. Table 3.2 shows the number of the operations executed by each entity in each phase of SPOON, respectively.

We also conduct an experiment to show the efficiency of SPOON. The operations of TA and service provider are performed on a notebook with Intel Core i5-4200U CPU, the clock rate is 2.29GHz and the memory is 4.00 GB. The operations of customers and mobile users are run on HUAWEI MT2-L01 smartphone with Kirin 910 CPU and 1250M memory. The operation system is Android 4.2.2 and the toolset is Android NDK r8d. We use MIRACL library 5.6.1 to implement number-theoretic based methods of cryptography. The Weil pairing is utilized to realize the bilinear pairing. The parameter p is approximately 160 bits and the elliptic curve is defined as $y = x^3 + 1$ over $\mathbb{F}_q$, where q is 512 bits. The execution time of each entity in every phase of SPOON is shown in Table 3.2. The running time is less than 300 ms for each entity. Therefore, our SPOON is quite efficient to be deployed on mobile devices.

3.5.2 Communication Overhead

We show the communication burden of all entities in SPOON. The public parameters are set the same as those in the experiment, that is, $|p|$=160 bits and $|q|$=512 bits. In user registration phase, a registrant, either customer or mobile user, sends a registering request

$(I, C, C', \widehat{A}, \mathcal{PK}_1)$ to the TA, which is $|I| + 2176$ bits, where $|I|$ is the binary length of the identity, and the TA returns $(A, B, s'', t'', e, f, P_0, RK)$ to the registrant, whose binary length is $|P_0| + 2176$ bits, where $|P_0|$ is the binary length of credit point. In task allocation, the customer uploads $(c_1, c_2, c_3, expires, \widehat{N}_{n\times n}, \gamma, w, \mathcal{PK}_2)$ and the mobile user sends $(\mu, \widetilde{N}_{n\times n}, \mathcal{PK}_3)$ to the service provider, which are $4512+160n^2+|expires|+|\gamma|+|w|$ bits and $2976 + 160n^2$ bits, respectively. The service provider responds $(num, c_2, c_3, c_4, expires, \gamma)$, which is $2560 + |num| + |expires| + |\gamma|$ bits, to a matched mobile user or false, 1 bit, to an unmatched one. After the mobile user obtains the sensing data, it generates the sensing report $(num, D_i, D_i', C_i', X_i, Y_i, Z_i, Q_i, \tau_j, \mathcal{SPK})$ to the service provider, which is $8864+|num|+|P_0|+|\tau_j|$ bits. The service provider needs to send 1024-bit W to the TA if a mobile user double-submits data, and then sends w sensing reports $(num, D_i, D_i', Y_i, Q_i, \tau_j)$, which is of binary length $w * (2560 + |num| + |P_0| + |\tau_j|)$ to the customer. Finally, the customer returns $(1024 + |\gamma|)$-bit (ϵ_i, Y_i) to the service provider for each report and the service provider sends $(B_i, t_i'', f_i, \theta_i, Y_i)$ to every mobile user, which is $1856 + |P_0|$ binary bits.

3.5.3 Credit Analysis

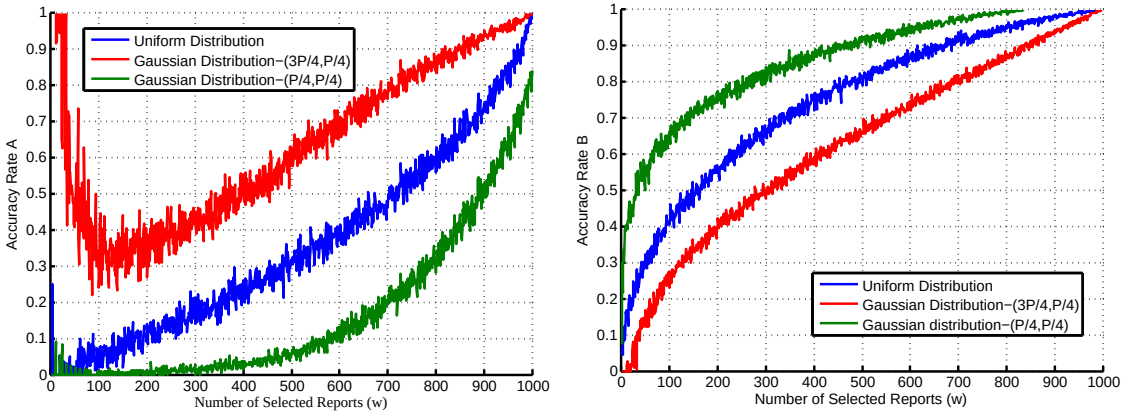
To prevent credit points from disclosing to other entities, each mobile user claims a threshold Q_i, which is less than its exact credit point P_i, such that the service provider can select the sensing reports based on the claimed thresholds. In this way, neither the service provider nor the customer enables to learn the precise credit points of mobile users. Unfortunately, this method reduces the accuracy of report selection as the service provider may select a sensing report of the mobile user, whose threshold is larger than others', while the credit point has the opposite result. On the other hand, customers may prefer mobile users to choose the thresholds that are approximate to their credit points, while the privacy of mobile users are sacrificed. Therefore, it seems impossible to reconcile the contradiction between privacy and accuracy, because they exhibit opposite trends.

To balance this trade-off, it is critical to find a reasonable strategy for mobile users to determine the thresholds. We define four parameters to evaluate privacy and accuracy in credit claiming. Specifically, accuracy rate A denotes the maximum probability of a given threshold in the selected reports can possess top-w credit points in sensing reports. Accuracy rate B denotes the maximum probability that a given credit point in the sensing reports is larger than the minimum threshold in the selected reports. Privacy rate A means the probability that a given sensing report, whose credit point is larger than the minimum of thresholds in selected reports, has top-w credit point in all sensing reports. Privacy rate B means the probability that a given sensing report would be selected by the service

provider, whose credit point is larger than the minimum of thresholds in selected reports. To determine how the threshold choosing strategy impacts the defined privacy and accuracy rates, we simulate the credit points of mobile users on Matlab and use different threshold choosing strategies to compute the accuracy rates and the privacy rates. The simulation results are illustrated in Fig. 3.3 and Fig. 3.4. We set the number of the mobile users to be 1000 in Fig. 3.4 and the number of the selected reports to be 100 in Fig. 3.4. We compare three threshold choosing strategies, the first one is basing on uniform distribution; the second one is based on Gaussian distribution, in which the mean is three quarters of credit points and the standard deviation is one quarter; the last one is based on Gaussian distribution, where the mean and the standard deviation are one quarter of credit points. The second strategy can achieve the highest accuracy and the third strategy can achieve the best privacy preservation on credit points in three strategies.

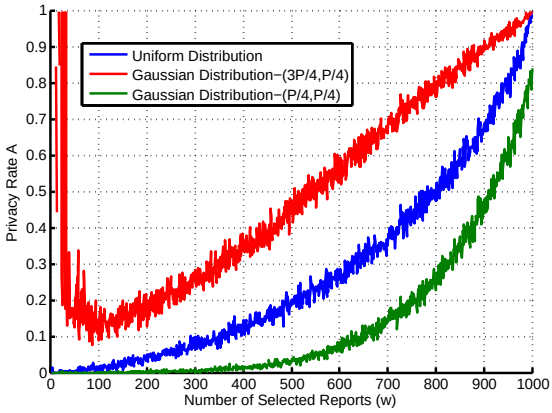
3.6 Summary

In this chapter, we have proposed a strong privacy-preserving mobile crowdsensing scheme with credit management to balance the trade-off between privacy preservation and task allocation. The service provider is allowed to select mobile users to perform sensing tasks according to the sensing areas of tasks and the geographic locations of mobile users, and select the sensing reports based on the credit points of mobile users. The sensitive information, including identities, locations, credit points, sensing tasks and sensing reports are preserved for mobile users and customers during task allocation and report selection. Furthermore, no trusted third party is required to achieve the credit management for mobile users. Finally, we have evaluated the security and privacy properties of the proposed scheme and demonstrated the scheme is sufficiently efficient to be implemented in real world.

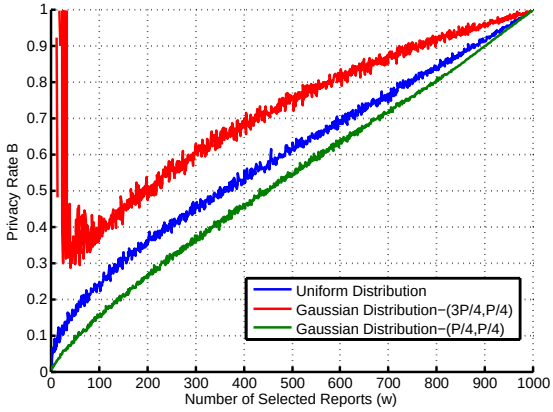


(a) Accuracy Rate A

(b) Accuracy Rate B

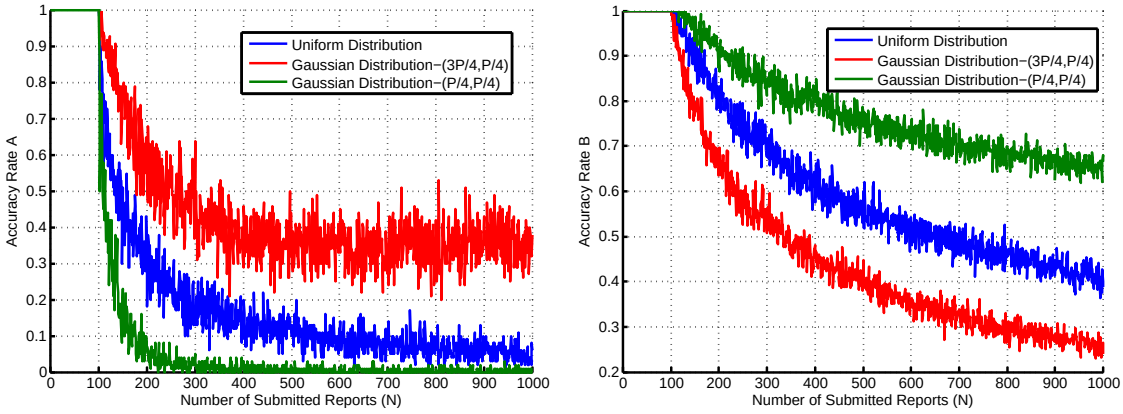


(c) Privacy Rate A



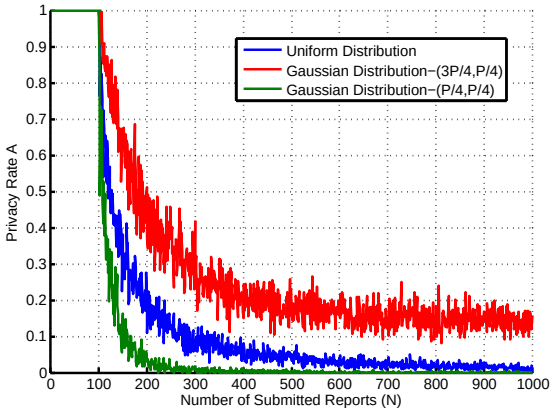
(d) Privacy Rate B

Figure 3.3: Accuracy and Privacy Rates with N=1000

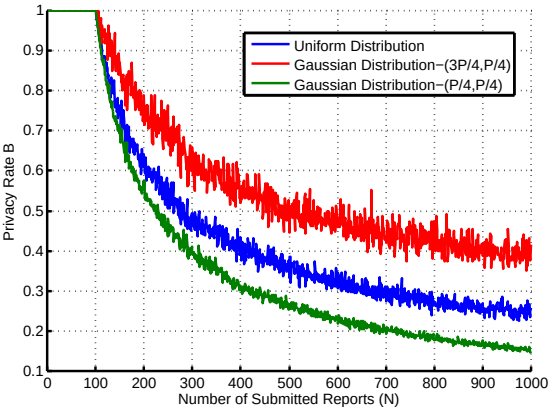


(a) Accuracy Rate A

(b) Accuracy Rate B



(c) Privacy Rate A



(d) Privacy Rate B

Figure 3.4: Accuracy and Privacy Rates with w=100

Chapter 4

Fog-assisted Secure Data Deduplication

4.1 Introduction

Mobile crowdsensing [1] is a compelling paradigm that allows a large group of individuals to collaboratively sense data and extract information about social events and national phenomena with common interest using their mobile devices, e.g., smart phones, smart glasses, drones, cameras and smart vehicles. It supports an ever-increasing number of sensing applications [116], ranging from social recommendation, such as restaurant recommendation, vehicular navigation and parking space discovery, to environment monitoring, such as air quality measurement, noise level measurement and dam water release warning [117, 118]. With the human intelligence and user mobility, it improves the quality of sensing data, extends the scale of sensing applications, and reduces the cost on high-quality data collection.

In mobile crowdsensing, one of the main challenges is to find proper mobile users for sensing tasks to achieve efficient and scalable data collection [112]. Firstly, due to the unique requirements of sensing tasks and the user mobility, a crowdsensing server (CS-server) collects various types of information about mobile users, e.g., location, reputation and activity pattern, and thereby customizes a task allocation policy for each sensing task [119]. For example, to measure traffic congestion in downtown Toronto, the CS-server should recruit the mobile users driving on the roads in downtown Toronto. Secondly, it is hard to guarantee that the potential mobile users could receive the assigned sensing tasks and upload sensing reports in time [120]. Thirdly, to perform sensing tasks, mobile

users have to travel to specific locations with a certain cost on time and travel. Therefore, there should be an effective framework for the CS-server to allocate sensing tasks to proper mobile users.

In addition, with the increasing number of participating mobile users, there are inevitably some duplicates in sensing reports [80]. For a social event or national phenomenon, mobile users in the same location may obtain the same sensing data and generate the identical or similar items in sensing reports. For example, let us consider the following two scenarios:

- ▷ To measure the air quality in an urban area, mobile users measuring at proximate points may submit the same measurements;
- ▷ To survey the satisfaction on the government in a certain area, some mobile users may make the same options and give similar comments.

In these cases, the replicate data consumes a large amount of network bandwidth and storage space. A straightforward method to reduce the overhead is to discard the redundant copies on intermediates. Nevertheless, this approach exposes the detailed sensing data, which may contain plenty of personal information about mobile users, e.g., location, references, occupations, health status and religious beliefs [94]. To preserve the privacy of mobile users, data encryption is widely used to achieve data confidentiality, but brings a huge obstacle on the detection of replicate data for intermediates. Message-locked encryption (MLE) [121] (the most prominent manifestation of which is convergent encryption) may resolve this problem, where the same plaintexts always map to the same ciphertexts. Nonetheless, MLE is inherently subject to off-line brute-force attacks [122], where adversaries can learn the sensing data by guessing the possible plaintexts in encrypted sensing reports. Furthermore, the sensing data is predictable in some mobile crowdsensing applications, such as air quality measurement, place recommendation and traffic congestion monitoring. As a result, an attacker may guess to obtain the correct sensing data in an encrypted sensing report. Therefore, it is necessary to design a secure data deduplication mechanism to allow the intermediates to detect replicate reports without violating the privacy of mobile users.

However, if the equality of sensing reports can be detected in public, anyone can predict that the mobile uses are in approximate positions or have similar preferences knowing that they submit the identical sensing reports. For this "duplicate-linking" leakage, some sensitive information would be exposed in duplicate-sensitive mobile crowdsensing applications, e.g., air quality monitoring and place recommendation. In these applications, the mobile

users in the same location or with the same profiles may report the identical sensing data. If two mobile users report identical measurements, the one may predict that the other is nearby. If two patients report the same symptoms, they may have the same disease [123]. Therefore, it is important to prevent privacy disclosure from the identical reports in mobile crowdsensing. However, data deduplication and privacy preservation are inherently in conflict. On one hand, if the intermediates can detect the replicate reports, they can know that there may be some correlation between the corresponding mobile users. On the other hand, if the privacy of mobile users is perfectly preserved, it is impossible for the intermediates to perform data deduplication. Therefore, it is an open problem to design a straightforward approach to prevent "duplicate-linking" leakage in duplicate-sensitive applications.

Furthermore, once the redundant copies in sensing reports are deleted, the CS-server cannot identify the contributions of mobile users. Although the redundant copies do not contribute to the completeness of sensing results, they improve their trustworthiness. The CS-server should not ignore the contributions of the mobile users whose data are replicate with others'. However, if these mobile users are rewarded, some lazy mobile users may acquire unfair benefits by replaying the sensing reports generated by other mobile users and eavesdropped on communication channels. This "duplicate-replay" attack should be abandoned to get rid of the lazy mobile users, who are unwilling to perform crowdsensing tasks, but greedy for benefits. In short, it is of significant importance to not only support secure data deduplication over sensing reports against brute-force attacks, "duplicate-linking" leakage and "duplicate-replay" attacks, but also record the contributions of mobile users fairly without exposing the sensing data.

To the end, we exploit fog computing [124] to support accurate task allocation and secure data deduplication for mobile crowdsensing, which is a new architecture providing computing, storage and networking services proximate to terminal devices with appealing properties, including location awareness, geographic distribution and low latency. With fog computing, a large number of decentralized mobile devices can self-organize to communicate and potentially collaborate with each other via a fog node located at the edge of the Internet [125]. In this chapter, we propose a Fog-assisted Mobile CrowdSensing framework (Fo-MCS) [126] that enables fog nodes to perform task allocation based on the mobility patterns of users to improve the accuracy of task assignment. Under this framework, we design a Fog-assisted Secure Data Deduplication scheme (Fo-SDD) based on BLS-Oblivious Pseudo-Random Function (BLS-OPRF) to achieve the detection of replicate data, and extend the Fo-SDD to prevent "duplicate-linking" leakage for duplicate-sensitive applications. Specifically, the main contributions of this chapter are as follows:

- ▷ We develop fog-assisted task allocation based on the local information about mobile users, such as mobility patterns and preferences. Specifically, the CS-server firstly assigns the sensing tasks to the fog nodes located in the intended sensing area. Then the fog nodes, acting as geography-related local servers, further find proper mobile users to fulfill the tasks. Fo-MCS not only reduces the overhead of CS-server on task allocation, but also improves the accuracy of task assignment.

- ▷ We design a BLS-OPRF scheme based on the BLS signature [127] to generate the encryption key of sensing data and enable fog nodes to detect and delete the identical sensing data in sensing reports for saving communication bandwidth. During this process, the fog nodes can learn nothing about the reports, except the equality of sensing data. Meanwhile, we also leverage the key-homomorphic signature [128] to sign the sensing data and allow fog nodes to aggregate the signatures of mobile users. By doing so, the CS-server only receives one copy of replicate sensing reports, but learns the contributions of the mobile users who generate these replicate sensing reports.

- ▷ We balance the trade-off between data deduplication and privacy preservation against "duplicate-linking" leakage. We leverage blind signatures [23] to ensure that no one can link the sensing reports to a specific mobile user, even if the user submits the identical reports with others. Nevertheless, once the participating mobile users are anonymous, it is difficult for the CS-server to distribute benefits based on their contributions. To address this issue, we utilize Chameleon hash function [129] to enable mobile users to claim their contributions and retrieve the corresponding rewards. In addition, the misbehavior of mobile users, including double-reporting of sensing data and double-retrieving of benefits, could be detected to guarantee the fairness of mobile crowdsensing.

4.2 Problem Statement

We formalize Fo-MCS framework and security threats. Then, we identify design goals.

4.2.1 Fo-MCS Framework

Fo-MCS framework consists of three layers: service layer, fog layer and mobile users layer, and four types of entities: customers, a CS-server, fog nodes and mobile users. In the service

layer, customers can be individuals or organizations. They have sensing tasks to fulfill but do not have sufficient resources to complete individually. Hence, they release these tasks on a CS-server. The CS-server provides mobile crowdsensing services for customers. It is responsible to assign sensing tasks to fog nodes based on the spatial information of tasks, process sensing reports, and distribute benefits to mobile users. In the fog layer, the fog nodes are deployed at the edge of the Internet and stretch from different network equipment, e.g., roadside units on roads, access points, gateways and edge routers. They have computing capability and storage space to provide computation and storage services to mobile users. Their responsibilities include assigning sensing tasks to mobile users on behalf of local servers, processing on sensing reports and forwarding the processed reports to the CS-server. In the mobile users layer, the mobile users perform the sensing tasks to collect data for earning rewards using their own mobile devices with the capabilities of data sensing, processing and communication.

As illustrated in Fig. 4.1, the whole Fo-MCS framework works as follows. A customer generates a sensing task and sends it to the CS-server, along with the rewards to attract mobile users. After obtaining the sensing task, the CS-server performs fog-assisted task allocation to assign it to mobile users. Specifically, the CS-server allocates the sensing task to the fog nodes according to the sensing area of the task and the coverage areas of fog nodes; and the fog nodes further recruit mobile users in their coverage areas to fulfill the task based on their mobility patterns and the task requirements. Then, the participating mobile users collect sensing data, generate sensing reports and submit them to the fog nodes. The fog nodes process the received sensing reports, including data deduplication and data aggregation, and forward the processed reports to the CS-server. After that, the CS-server generates a crowdsensing result for the customer based on the processed reports. Finally, the customer reads the crowdsensing result and determines the contributions of mobile users and the CS-server distributes the rewards to mobile users according to their contributions on the sensing task.

4.2.2 Threat Model

Security threats come from both external and internal attackers. The global eavesdroppers may wiretap on wireless communication channels to capture the messages exchanged between two entities, e.g., fog nodes and mobile devices. The CS-server and fog nodes are both honest-but-curious, indicating that they follow the protocols agreed with customers and mobile users honestly, but they are also interested in the sensing reports generated by mobile users. The mobile users are honest to perform sensing tasks for benefits, but curious on the sensing reports submitted by other users, lazy for sensing data and greedy

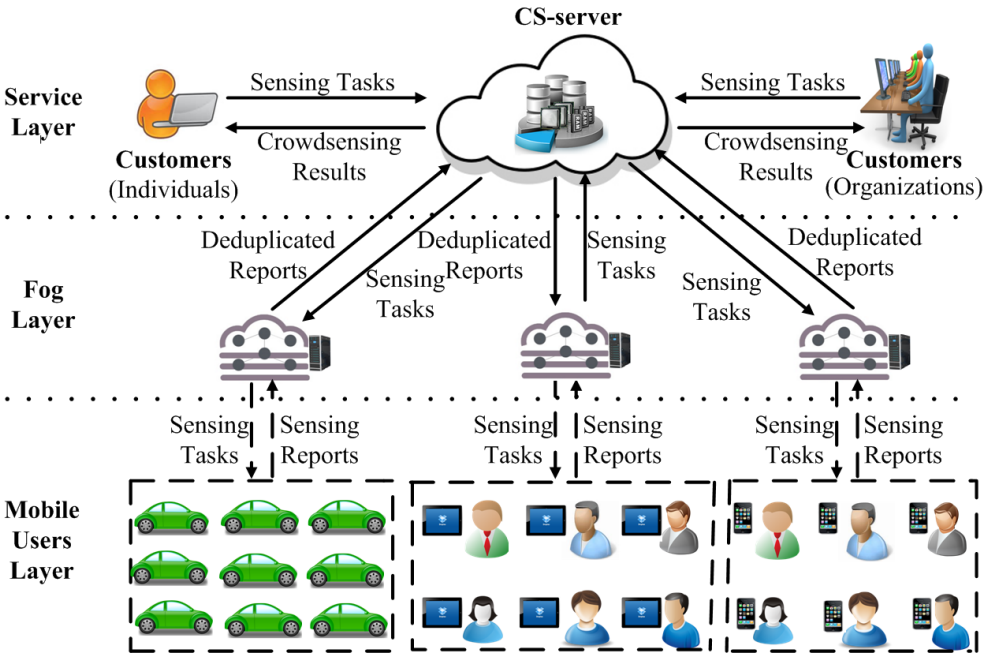


Figure 4.1: Fo-MCS Framework

for benefits. Specifically, the attackers may launch the following attacks to achieve their goals:

- ▷ Brute-Force Attack: A curious entity, including the CS-server or mobile users, checks all possible sensing data or measurements with the hope of eventually obtaining the correct plaintexts in the encrypted sensing reports.
- ▷ "Duplicate-Linking" Leakage: The identical sensing reports disclose the equality of sensing data generated by mobile users. Thus, it is predictable that these mobile uses are in proximate positions or have similar profiles, such as references, habits or health status.
- ▷ "Duplicate-Replay" Attack: A lazy mobile user captures a sensing report delivered by others through eavesdropping and replays it to cheat the fog node to believe that his report is identical with a submitted one. Thus, the mobile user would be rewarded although the replayed report will be deleted by the fog node.
- ▷ Double-Reporting: A greedy mobile user may submit more sensing reports than

allowed without being detected, in such a way that the user would obtain more rewards.

- ▷ Double-Retrieving: To acquire more benefits than that rewarded by the CS-server, a greedy mobile user may retrieve the rewards more than once without being detected.

In addition, the mobile users may deliver forged sensing data to cheat customers for benefits. This active attack has been discussed in [130, 131], which can be resisted by using trust management of mobile users or tasks duplication among multiple participants (to recruit multiple mobile users to collect the same data or measure the same phenomenon). Therefore, we assume that the majority of mobile users are fully trusted to perform the sensing tasks, and multiple fog nodes would not collude together, or collude with the CS-server to invade the privacy of mobile users. The customers are honest as they are the beneficiaries of mobile crowdsensing services.

4.2.3 Design Goals

To achieve secure data deduplication under the Fo-MCS framework and resist the security threats, Fo-SDD should achieve the following design goals.

- ▷ *Secure Data Deduplication*: To save communication bandwidth, the replicate data in sensing reports should be securely deleted. Specifically, the fog nodes are able to detect and erase the replicate data without learning any information about the sensing reports. To ensure the confidentiality of sensing data, Fo-SDD should satisfy the following security goals:
  - Security against Brute-Force Attacks: The sensing data should be encrypted to prevent attackers from recovering it through brute-force attacks. Although the semantic security cannot be achieved, Fo-SDD should reach high security guarantee, except that the encrypted sensing reports expose the equality of underlying sensing data.
  - No "Duplicate-Linking" Leakage: The privacy leakage from the equality of sensing data should be prevented in duplicate-sensitive applications. A mobile user cannot predict that another user is similar with him in some aspects if they submit identical sensing reports.

- Security against "Duplicate-Replay" Attacks: To prevent lazy mobile users from replaying the captured sensing reports, it is necessary to prove that the mobile users actually possess the sensing data if they submit the corresponding reports to fog nodes. A lazy mobile user can be detected if he replays captured sensing reports generated by others.

▷ *Efficient Contribution Claim*: The contributions of mobile users who submit the replicate sensing reports should not be ignored. To reduce communication overhead and record the contributions of mobile users, the fog nodes are able to aggregate the signatures on the identical sensing data generated by different mobile users. In addition, to maintain the fairness of mobile users, Fo-SDD should achieve the following goals:

- Detection of Double-Reporting: A greedy mobile user cannot submit more sensing reports than allowed to the CS-server without being detected.
- Detection of Double-Retrieving: A greedy mobile user cannot double-retrieve the rewards from the CS-server without being detected.

In addition, to offer sophisticated security protection on the Fo-MCS framework, we should achieve other fundamental security goals, such as the confidentiality of sensing tasks against external attackers, the authentication and integrity of sensing reports.

4.3 Fo-SDD

We introduce the overview of Fo-SDD, describe the Fo-SDD in detail and discuss the security properties of Fo-SDD.

4.3.1 High-Level Description

To resist brute-force attacks, we design a BLS-OPRF scheme to prevent attackers from guessing the predictable sensing data. Specifically, the local fog node $\mathcal{F}_j$ aids each mobile user $\mathcal{U}_i$ in its coverage area to generate the encryption key $\mathcal{S}_i$ with its secret key x_j for the sensing data $\mathcal{P}_i$. Thus, the attackers cannot compute $\mathcal{S}_i$ without x_j and thereby recover $\mathcal{P}_i$ from the target ciphertext Z_i using brute-force attacks. Unfortunately, a curious $\mathcal{F}_j$ is still able to guess $\mathcal{P}_i$ in Z_i. To prevent the brute-force attacks of $\mathcal{F}_j$, we employ mobile fog nodes to generate the encryption key $\mathcal{S}_i$ of sensing data $\mathcal{P}_i$ for $\mathcal{U}_i$. As a result, a single fog

node $\mathcal{F}_j$ cannot launch brute-force attacks to recover $\mathcal{P}_i$ from Z_i. In addition, to further reduce communication overhead and achieve contribution claim, we allow each mobile user to generate a signature on the sensing data based on the key-homomorphic signature [128]. The distinguished feature of the key-homomorphic signature is that the signatures from multiple mobile users who generate replicate reports can be aggregated to be one signature, while all the public keys of these mobile users should be used to verify the validity of the aggregated signature. In this way, the communication overhead between fog nodes and CS-server is reduced and the customer can learn the identities of contributors during the verification of the aggregated signature. In addition, proxy re-encryption [22] is leveraged to realize the confidentiality of sensing tasks and allow the CS-server to efficiently delegate the decryption capability of sensing tasks to fog nodes.

4.3.2 The Detailed Fo-SDD

Fo-SDD consists of six phases: Service-Setup, Task-Releasing, Task-Allocation, Data-Collection, Data-Deduplication and Data-Reading. The detailed Fo-SDD is described below.

4.3.2.1 Service-Setup

This phase is run by the CS-server to bootstrap mobile crowdsensing services. Given the security parameter λ, the CS-server defines two cyclic groups $(\mathbb{G}, \mathbb{G}_T)$ with the same prime order p, where p is λ bits. Let g be a random generator of group $\mathbb{G}$, $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ be a bilinear map. $H : \{0,1\}^* \to \mathbb{G}$ and $\mathcal{H} : \{0,1\}^* \to \{0,1\}^\lambda$ are two full-domain hash functions, such as SHA-256 or SHA-3 [20]. $(\mathcal{SE}, \mathcal{SD})$ are the encryption and decryption algorithms of a deterministic symmetric encryption scheme. Such a scheme can be constructed from AES scheme with a fixed IV in CTR mode [20]. (SE, SD) are the encryption and decryption algorithms of the standard AES scheme [20]. The CS-server randomly chooses $s \in \mathbb{Z}_p$ as its secret key and computes $S = g^s \in \mathbb{G}$ as its public key. The DSA signature [20] is employed to achieve the integrity and authentication of sensing tasks and sensing reports during transmission.

A fog node $\mathcal{F}_j$ randomly chooses $x_j \in \mathbb{Z}_p$ as the secret key and computes $X_j = g^{x_j} \in \mathbb{G}$ as the public key.

A mobile user $\mathcal{U}_i$ randomly picks $v_i \in \mathbb{Z}_p$ as the secret key and computes $U_i = \hat{e}(g,g)^{v_i}$ as the public key.

4.3.2.2 Task-Releasing

When a customer $\mathcal{C}$ is willing to collect data for some purpose, such as measuring air quality, monitoring traffic condition or reconstructing indoor floor plan, $\mathcal{C}$ first generates a sensing task $\mathcal{T} = (\mathcal{T}_t, \mathcal{T}_e, \mathcal{T}_a, \mathcal{T}_b)$, indicating the goal (what to sense), the expiration time (when to sense), the sensing area (where to sense) and the benefits, respectively. Then, $\mathcal{C}$ chooses a random $k \in \mathbb{Z}_p$ to calculate a temporary public key $K = g^k$. After that, $\mathcal{C}$ encrypts $\mathcal{T}$ by randomly picking $r \in \mathbb{Z}_p$, $T \in \mathbb{G}_T$ to compute

$$(C_1, C_2, C_3) = (S^r, T\hat{e}(g,g)^r, SE(\mathcal{H}(C_2||T), \mathcal{T}_t||\mathcal{T}_e||\mathcal{T}_b)).$$

Finally, $\mathcal{C}$ sets $C_c = (C_1, C_2, C_3)$ and sends $(C_c, K, \mathcal{T}_a)$ to the CS-server.

4.3.2.3 Task-Allocation

Upon receiving $(C_c, K, \mathcal{T}_a)$, the CS-server first chooses $\mathcal{N} \in \mathbb{Z}_p$ as a unique identifier of $\mathcal{T}$ and picks a set of fog nodes $\mathbb{F} = \{\mathcal{F}_1, \cdots, \mathcal{F}_N\}$ located in $\mathcal{T}_a$, where N is the number of fog nodes in the set $\mathbb{F}$. Then, for each $\mathcal{F}_j \in \mathbb{F}$, the CS-server uses s and X_j to compute

$$RK_j = X_j^{\frac{1}{s}}, \qquad C'_j = \hat{e}(C_1, RK_j).$$

Finally, the CS-server sends $(\mathcal{N}, C'_j, C_2, C_3, K, \mathcal{T}_a)$ to $\mathcal{F}_j$.

When $\mathcal{F}_j$ receives $(\mathcal{N}, C'_j, C_2, C_3, K, \mathcal{T}_a)$, it first decrypts the sensing task as $T' = \frac{C_2}{(C'_j)^{1/x_j}}$ and $\mathcal{T}_t||\mathcal{T}_e||\mathcal{T}_b = SD(\mathcal{H}(C_2||T'), C_3)$. Then, $\mathcal{F}_j$ checks whether the task $\mathcal{T}$ is expired or not. If not, $\mathcal{F}_j$ recruits a set of mobile users $\mathbb{U} = \{\mathcal{U}_1, \cdots, \mathcal{U}_M\}$ to perform $\mathcal{T}$ based on the requirements of $\mathcal{T}$ and the mobility patterns of mobile users [112, 55, 59], where M is the number of mobile users in $\mathbb{U}$. For each $\mathcal{U}_i \in \mathbb{U}$, $\mathcal{F}_j$ randomly chooses $r_i \in \mathbb{Z}_p$, $R \in \mathbb{G}_T$ and encrypts $\mathcal{T}$ as

$$(D_{i1}, D_{i2}, D_{i3}) = (g^{r_i}, RU_i^{r_i}, SE(\mathcal{H}(\mathcal{N}||R), \mathcal{T}_t||\mathcal{T}_e||\mathcal{T}_b)).$$

Finally, $\mathcal{F}_j$ sets $D_i = (D_{i1}, D_{i2}, D_{i3})$ and sends $(\mathcal{N}, D_i, K, \mathcal{T}_a)$ to $\mathcal{U}_i$.

4.3.2.4 Data-Collection

When receiving $(\mathcal{N}, D_i, K, \mathcal{T}_a)$, $\mathcal{U}_i$ first decrypts the ciphertext D_i as $R' = \frac{D_{i2}}{\hat{e}(g,D_{i1})^{v_i}}$ and $\mathcal{T}_t||\mathcal{T}_e||\mathcal{T}_b = SD(\mathcal{H}(\mathcal{N}||R'), D_{i3})$. If this task is not expired, $\mathcal{U}_i$ starts to perform the task, collect and generate the sensing data $\mathcal{P}_i$ according to the requirements of $\mathcal{T}$. Then, $\mathcal{U}_i$ randomly chooses $s_i \in \mathbb{Z}_p$ to compute $S_i = H(\mathcal{P}_i)^{s_i}$ and sends $(\mathcal{N}, S_i)$ to $\mathcal{F}_j$. After $\mathcal{F}_j$ receives $(\mathcal{N}, S_i)$, it calculates $S_i' = S_i^{x_j}$ and returns S_i' to $\mathcal{U}_i$ (To prevent brute-force attacks from $\mathcal{F}_j$, S_i' can be generated by multiple fog nodes, that is, $S_i' = S_i^{\sum_{j\in\mathcal{M}} x_j}$, where $\mathcal{M}$ is the set of indices of $\mathcal{F}_j$ and its neighboring fog nodes). Then, $\mathcal{U}_i$ computes $\mathcal{S}_i = (S_i')^{\frac{1}{s_i}}$ and verifies whether

$$\hat{e}(H(\mathcal{P}_i), X_j) = \hat{e}(\mathcal{S}_i, g) \tag{4.1}$$

or not. If not, $\mathcal{U}_i$ returns failure and aborts; otherwise, it computes $Z_i = \mathcal{SE}(\mathcal{H}(\mathcal{N}||\mathcal{S}_i), \mathcal{P}_i)$. Furthermore, $\mathcal{U}_i$ chooses a random $w_i \in \mathbb{Z}_p$ and generates a signature σ_i as

$$\sigma_i = (\sigma_{i1}, \sigma_{i2}) = (g^{-w_i}, g^{v_i} H(\mathcal{N}||\mathcal{S}_i, \mathcal{P}_i)^{w_i}).$$

$\mathcal{U}_i$ sets the sensing report $\mathbb{P}_i = (Z_i, \sigma_i)$ and sends $(\mathcal{N}, \mathbb{P}_i)$ to $\mathcal{F}_j$. Upon receiving $(\mathcal{N}, \mathbb{P}_i)$, $\mathcal{F}_j$ computes $Y_i = \mathcal{H}(\mathcal{N}||Z_i)$ and checks whether Y_i exists in the database. If yes, $\mathcal{F}_j$ returns success and aborts; otherwise, it keeps $(\mathcal{N}, \mathbb{P}_i, Y_i)$ and requests $\mathcal{U}_i$ to return (W_i, J_i). $\mathcal{U}_i$ generates (W_i, J_i) by picking a random number $a_i \in \mathbb{Z}_p$ to calculate $W_i = g^{a_i}$, $a_i' = \mathcal{H}(W_i||K^{a_i})$, $J_i = SE(a_i', \mathcal{S}_i)$ and sends (W_i, J_i) to $\mathcal{F}_j$.

It is worth pointing out that the ciphertext of sensing data $\mathcal{P}_i$ supports replicate data detection and deletion. Specifically, Y_i is a tag used to detect the duplicate of $\mathcal{P}_i$. $\mathcal{S}_i$ is derived from $(\mathcal{N}, \mathcal{P}_i)$ and used to encrypt $\mathcal{P}_i$. Therefore, $\mathcal{F}_j$ is able to detect the replicate data based on Y_i in sensing reports. The data collection and deduplication processes are shown in Fig. 4.2, and the information flow of data collection phase is illustrated in Fig. 4.3.

4.3.2.5 Data-Deduplication

Upon receiving $\{\mathbb{P}_1, \cdots, \mathbb{P}_M\}$ from $\mathbb{U}$, $\mathcal{F}_j$ checks whether $\{\mathbb{P}_1, \cdots, \mathbb{P}_M\}$ are replicate or not. If there are two reports, in which Y_i=$Y_{\hat{i}}$, the reports in $\mathbb{P}_i$ and $\mathbb{P}_{\hat{i}}$ are identical. If a set of reports $\{\mathbb{P}_i\}_{i\in Q}$ are identical, where Q is the set of indices of replicate reports, $\mathcal{F}_j$ aggregates the corresponding signatures $\{\sigma_i\}_{i\in Q}$ as

$$\sigma_Q = (\sigma_{Q1}, \sigma_{Q2}) = (\textstyle\prod_{i\in Q} \sigma_{i1}, \prod_{i\in Q} \sigma_{i2}).$$

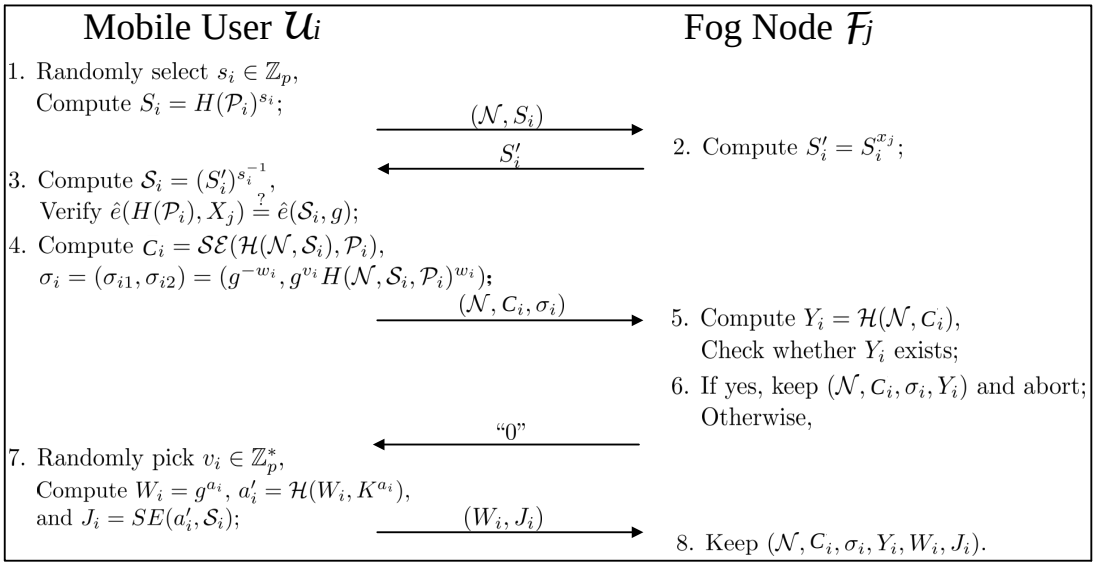


Figure 4.2: Information Flow of Data Collection

Then, $\mathcal{F}_j$ keeps the first copy $Z_{\hat{i}}$ generated by $\mathcal{U}_{\hat{i}}$ who delivers $(W_{\hat{i}}, J_{\hat{i}})$ and deletes the replicate copies. $\mathcal{F}_j$ sets the sensing reports that are not replicate with others as $\{\mathbb{P}_i\}$ for each $1 \leq i \leq M$ and $i \notin Q$. Finally, $\mathcal{F}_j$ forwards the deduplicated reports $(\mathcal{N}, \{(Z_i, \sigma_i, W_i, J_i)\}_{i \notin Q}, Z_{\hat{i}}, \sigma_Q, W_{\hat{i}}, J_{\hat{i}})$ to the CS-server.

When receiving $(\mathcal{N}, \{(\widehat{\mathbb{P}}_i, W_i, J_i)\}_{i \notin Q}, Z_{\hat{i}}, \sigma_Q, W_{\hat{i}}, J_{\hat{i}})$ from $\mathcal{F}_j$, the CS-server forwards them to the customer $\mathcal{C}$.

4.3.2.6 Data-Reading

When $\mathcal{C}$ receives the deduplicated reports from the CS-server, $\mathcal{C}$ uses k to decrypt the deduplicated reports and checks the contributors (mobile users) as follows:

▷ For each $(Z_i, \sigma_i, W_i, J_i) \in \{(Z_i, \sigma_i, W_i, J_i)\}_{i \notin Q}$, $\mathcal{C}$ computes

$$\mathcal{S}_i = SD(\mathcal{H}(W_i || W_i^k), J_i), \quad \mathcal{P}_i = \mathcal{SD}(\mathcal{H}(\mathcal{N} || \mathcal{S}_i) || Z_i).$$

After recovering all sensing reports $\{(Z_i, \sigma_i, W_i, J_i)\}_{i \notin Q}$ that are not replicate with others, $\mathcal{C}$ verifies the signatures $\{\sigma_i\}_{i \notin Q}$ by checking whether

$$\hat{e}(\prod_{i \notin Q} \sigma_{i2}, g) \prod_{i \notin Q} \hat{e}(H(\mathcal{N} || \mathcal{S}_i || P_i), \sigma_{i1}) \stackrel{?}{=} \prod_{i \notin Q} U_i. \tag{4.2}$$

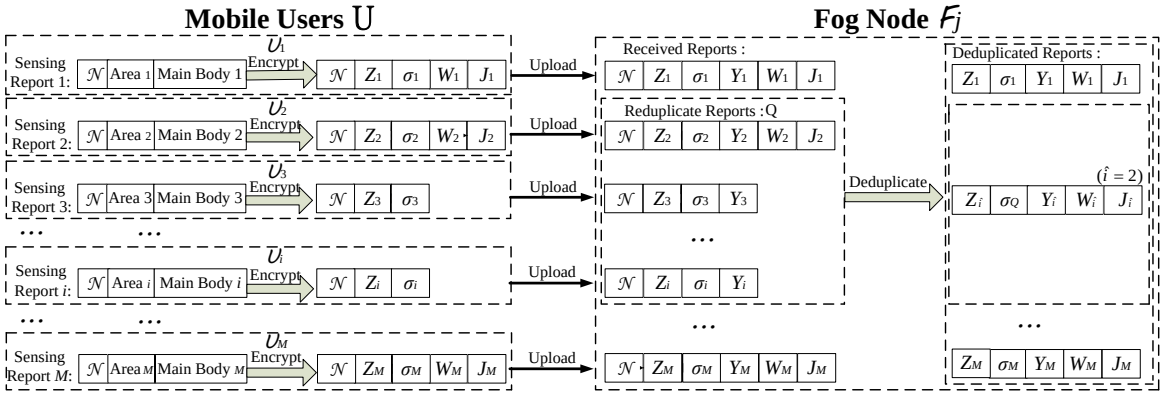


Figure 4.3: Data Collection and Deduplication

If yes, $\mathcal{C}$ accepts the sensing data $\{\mathcal{P}_i\}_{i\notin Q}$ and learns $\{\mathcal{U}_i\}_{i\notin Q}$ are the contributors; otherwise, $\mathcal{C}$ uses the recursive divide-and-conquer approach to find and delete the corrupted data.

▷ For $(Z_{\hat{i}}, \sigma_Q, W_{\hat{i}}, J_{\hat{i}})$, $\mathcal{C}$ computes

$$\mathcal{S}_{\hat{i}} = SD(\mathcal{H}(W_{\hat{i}}||W_{\hat{i}}^k)||J_{\hat{i}}), \quad \mathcal{P}_{\hat{i}} = \mathcal{SD}(\mathcal{H}(\mathcal{N}||\mathcal{S}_{\hat{i}})||Z_{\hat{i}}).$$

After obtaining the sensing data $\mathcal{P}_{\hat{i}}$, $\mathcal{C}$ verifies the signature σ_Q by checking whether

$$\hat{e}(\sigma_{Q2}, g)\hat{e}(H(\mathcal{N}||\mathcal{S}_{\hat{i}}||P_{\hat{i}}), \sigma_{Q1}) \stackrel{?}{=} \prod_{i\in Q} U_i. \tag{4.3}$$

If yes, $\mathcal{C}$ accepts the sensing data $\mathcal{P}_{\hat{i}}$ and learns $\{\mathcal{U}_i\}_{i\in Q}$ are the contributors of $\mathcal{P}_{\hat{i}}$; otherwise, $\mathcal{C}$ deletes it.

Finally, $\mathcal{C}$ obtains the sensing data $(\{\mathcal{P}_i\}_{i\notin Q}, \mathcal{P}_{\hat{i}})$ and distributes the benefits to mobile users in $\mathbb{U}$ based on their contributions.

4.3.3 Security Analysis

We demonstrate the achievement of security properties, i.e., secure data deduplication and efficient contribution claim. *Secure Data Deduplication*: To deduplicate sensing reports and achieve data confidentiality, a BLS-OPRF scheme is designed to compute the encryption

key from the sensing data. Thus, a fog node is able to detect the replicate data based on the ciphertexts, which are identical if the sensing data is equal. The Fo-SDD not only supports the deduplication of sensing reports, but also achieves high security guarantee on sensing data.

- ▷ Security against Brute-Force Attacks: The encryption key $\mathcal{S}_i$ is secret that no adversary is able to distinguish it from a random value, except $\mathcal{F}_j$. With $(H(\mathcal{P}_i)^{s_i}, H(\mathcal{P}_i)^{s_i x_j})$, it is hard to compute $H(\mathcal{P}_i)^{x_j}$; otherwise, the Computational Diffie-Hellman (CDH) problem [127] is intractable. Moreover, since the external attackers do not have the secret key of $\mathcal{F}_j$, they cannot guess the sensing data $\mathcal{P}_i$. Thus, the sensing data is confidential against brute-force attacks. Nonetheless, since the Decisional Diffie-Hellman (DDH) problem [127] is tractable, it is possible for $\mathcal{F}_j$ to obtain $\mathcal{P}_i$ by using brute-force attacks, that is, to guess a $\mathcal{P}_i'$ and test whether $\hat{e}(H(\mathcal{P}_i'), S_i') = \hat{e}(S_i, \mathcal{S}_i)$ holds or not. To prevent this attack from $\mathcal{F}_j$, we employ multiple neighboring fog nodes to cooperatively generate S_i' for $\mathcal{U}_i$. Thereby, a single fog node $\mathcal{F}_j$ cannot launch brute-force attacks to acquire $\mathcal{P}_i$, unless all neighboring fog nodes collude to guess. Certainly, to achieve higher security guarantee against brute-force attacks, it is possible to employ a trusted key server to generate encryption keys for all mobile users, such as a cellular service provider or network operator.
- ▷ Security against "Duplicate-Replay" Attacks: To prevent lazy mobile users from replaying other users' sensing reports, we ensure that only the mobile users possessing sensing data can generate valid sensing reports. Specifically, $\mathcal{U}_i$ needs to use the sensing data $\mathcal{P}_i$ to generate the signature σ_i, which would be verified by the customer in Data-Reading phase. If $\mathcal{U}_i$ replays a captured report Z_i without possessing $\mathcal{P}_i$, the misbehavior can be detected by the customer. Therefore, as long as the key homomorphic signature [128] is unforgeable, Fo-SDD is secure against "duplicate-replay" attacks.

Efficient Contribution Claim: To claim the contribution, $\mathcal{U}_i$ utilizes the key-homomorphic signature scheme to generate σ_i on $\mathcal{P}_i$, such that $\mathcal{C}$ can confirm whether $\mathcal{U}_i$ is the contributor of $\mathcal{P}_i$ or not by verifying σ_i. If some mobile users $\{\mathcal{U}_i\}_{i\in Q}$ upload the same data $\mathcal{P}_{\hat{i}}$, $\mathcal{F}_j$ aggregates the corresponding signatures $\{\sigma_i\}_{i\in Q}$ to generate σ_Q for reducing the communication overhead from $\mathcal{F}_j$ to $\mathcal{C}$. Meanwhile, $\mathcal{C}$ is able to verify the validity of the signature σ_Q with the public keys of $\{\mathcal{U}_i\}_{i\in Q}$. Therefore, $\mathcal{C}$ approves that $\mathcal{P}_{\hat{i}}$ is generated by $\{\mathcal{U}_i\}_{i\in Q}$. Since the key-homomorphic signature [128] achieves existential unforgeability based on the CDH assumption, no attacker can forge the signatures or claim the contributions of the eligible mobile users to itself. Therefore, $\mathcal{C}$ believes that the claimed contributions of mobile users should belong to them indeed.

In addition, it is possible to detect double-reporting and double-retrieving, since both the customer and the CS-server know the identities of mobile users. The CS-server can record the identities when the mobile users deliver their reports and retrieve their rewards. Once a mobile user double-submits reports or double-retrieves rewards, the CS-server can find the misbehavior by checking the records.

4.4 Extended Fo-SDD

Since the symmetric encryption scheme used to encrypt sensing data in Fo-SDD is deterministic, it produces the same ciphertext from a given identical plaintext and an encryption key, when it is separately executed by different mobile users. If several mobile users generate the same sensing data $\mathcal{P}_i$, the ciphertexts are identical, that is, Z_i. As a result, any one can learn that two sensing reports are identical, and thereby predict that these mobile uses are in proximate positions or have similar profiles. To prevent privacy leakage of mobile users in duplicate-sensitive applications, we extend the Fo-SDD by means of anonymization. Specifically, we leverage the blind signature [23] to extend the Fo-SDD to prevent malicious hackers or curious entities from learning the identities of participating mobile users. Unfortunately, once the mobile users are anonymous, some problems are emerged. For example, greedy mobile users may submit more sensing reports than allowed to earn unfair benefits; it is difficult for the CS-server to distribute rewards to mobile users; and greedy mobile users may double-draw their rewards from the CS-server. Therefore, we design a contribution claim and reward retrieval mechanism from Chameleon hash function [129] to allow mobile users to claim their contributions and retrieve their rewards fairly. Meanwhile, the CS-server is able to discover the misbehavior of greedy mobile users, and thereby recover their identities.

4.4.1 Extended Fo-SDD

The detailed description of the extended Fo-SDD scheme is shown below.

4.4.1.1 Service-Setup

The CS-server bootstraps the mobile crowdsensing services following the same procedures as those in the Fo-SDD, except that five more public parameters $(g_0, g_1, G, \mathcal{G}, \mathcal{F})$ are needed.

g_0, g_1 are two random generators of group $\mathbb{G}$, G is a random value chosen from $\mathbb{G}_T$, $\mathcal{G} = \hat{e}(g, g)$ and $F : \mathbb{Z}_p \times \{0,1\}^* \rightarrow \mathbb{Z}_p$ is a pseudo-random function.

A fog node $\mathcal{F}_j$ randomly chooses $x_j \in \mathbb{Z}_p$ as the secret key and computes $X_j = g^{x_j} \in \mathbb{G}$ as the public key.

A mobile user $\mathcal{U}_i$ randomly picks $v_i \in \mathbb{Z}_p$ as the secret key and computes $U_i = \mathcal{G}^{v_i}$ as the public key. $\mathcal{U}_i$ is required to register at the CS-server to obtain an anonymous credential, which is used to access the crowdsensing services, in the following steps:

- ▷ $\mathcal{U}_i$ randomly chooses $u_i' \in \mathbb{Z}_p$ to compute $A_i = g_0^{u_i'} g_1^{v_i}$, and sends (A_i, U_i) to the CS-server, along with the following zero-knowledge proof expressed in Camenisch-Stalder notation [132]:

$$\mathcal{PK}\{(u_i', v_i) : A_i = g_0^{u_i'} g_1^{v_i} \wedge U_i = \mathcal{G}^{v_i}\}.$$

- ▷ The CS-server checks $\mathcal{PK}$ to ensure (A_i, U_i) is generated properly. It randomly picks $u_i'', e_i \in \mathbb{Z}_p$ to calculate $B_i = (gA_i g_0^{u_i''})^{\frac{1}{s+e_i}}$ and returns (B_i, u_i'', e_i) to $\mathcal{U}_i$.

- ▷ $\mathcal{U}_i$ computes $u_i = u_i' + u_i''$ and checks $\hat{e}(B_i, Sg^{e_i}) \stackrel{?}{=} \hat{e}(gg_0^{u_i} g_1^{v_i}, g)$. If yes, $\mathcal{U}_i$ maintains (B_i, e_i, u_i) along with v_i.

Finally, $\mathcal{U}_i$ obtains an anonymous credential (B_i, e_i, u_i).

4.4.1.2 Task-Releasing

The Task-Releasing phase is the same as that in Fo-SDD.

4.4.1.3 Task-Allocation

The Task-Allocation phase is the same as that in Fo-SDD.

4.4.1.4 Data-Collection

$\mathcal{U}_i$ follows the same operations as those in Fo-SDD to recover $\mathcal{T}_t||\mathcal{T}_e||\mathcal{T}_b$, generate $\mathcal{P}_i$, interact with the CS-server to compute $\mathcal{S}_i$ and encrypt $\mathcal{P}_i$ to obtain Z_i. Furthermore, $\mathcal{U}_i$ randomly chooses $b_i \in \mathbb{Z}_p$ to compute $Y_i = \mathcal{H}(\mathcal{N}||Z_i||t)$, $l_i = F(v_i, \mathcal{N}||t||U_i)$, $V_i = g_0^{l_i}$, $T_i = \mathcal{G}^{v_i} G^{Y_i l_i}$,

$H_i = \mathcal{G}^{l_i} {U_i}^{b_i}$, where t denotes the current reporting period. After that, $\mathcal{U}_i$ picks a random value $a_i \in \mathbb{Z}_p$ to calculate $W_i = g^{a_i}$, $a_i' = \mathcal{H}(W_i || K^{a_i})$, and $J_i = SE(a_i', \mathcal{S}_i)$. Finally, $\mathcal{U}_i$ sends the sensing report $\mathbb{P}_i = (\mathcal{N}, Z_i, V_i, T_i, H_i, W_i, J_i)$ to $\mathcal{F}_j$, along with the following zero-knowledge proof expressed in Camenisch-Stalder notation [132]:

$$\mathcal{SPK}\left\{\begin{array}{c}(B_i, e_i, u_i, v_i, l_i, b_i): \\ \hat{e}(B_i, Sg^{e_i}) = \hat{e}(gg_0^{u_i} g_1^{v_i}, g) \wedge \\ V_i = g_0^{l_i} \wedge \\ T_i = \mathcal{G}^{v_i} G^{Y_i l_i} \wedge \\ H_i = \mathcal{G}^{l_i} U_i^{b_i}\end{array}\right\}(Z_i).$$

4.4.1.5 Data-Deduplication

Upon receiving $\{\mathbb{P}_1, \cdots, \mathbb{P}_M\}$ from $\mathbb{U}$, $\mathcal{F}_j$ first verifies the validity of $\mathcal{SPK}$ and checks whether there are double-submitted reports. Given two sensing reports, $\mathbb{P}_i = (\mathcal{N}, Z_i, V_i, T_i, H_i, W_i, J_i)$ and $\mathbb{P}'_i = (\mathcal{N}, Z_i', V_i', T_i', H_i', W_i', J_i')$, $\mathcal{F}_j$ computes $Y_i = \mathcal{H}(\mathcal{N} || Z_i || t)$, $Y_i' = \mathcal{H}(\mathcal{N} || Z_i' || t)$. If $V_i = V_i'$ and $Y_i \neq Y_i'$, the mobile user double-submits two reports in a time period. $\mathcal{F}_j$ recovers the public key of the greedy mobile user by computing $U_i = (\frac{(T_i)^{Y_i'}}{(T_i')^{Y_i}})^{\frac{1}{Y_i' - Y_i}}$, and deletes one of the reports. If $V_i = V_i'$ and $Y_i = Y_i'$, these reports are the same and submitted by the same mobile user. $\mathcal{F}_j$ keeps one of them. If $Y_i = Y_{\hat{i}}$ and $V_i \neq V_i'$, two sensing reports $\mathbb{P}_i$ and $\mathbb{P}_{\hat{i}}$ are identical, but delivered by different mobile users. $\mathbb{P}_i$ and $\mathbb{P}_{\hat{i}}$ are replicate reports. If a set of reports $\{\mathbb{P}_i\}_{i \in Q}$ are replicate, $\mathcal{F}_j$ keeps the first received copy $(Z_{\hat{i}}, W_{\hat{i}}, J_{\hat{i}})$ generated by $\mathcal{U}_{\hat{i}}$ and deletes the replicate copies. The sensing reports that are not replicate with others are $\{\mathbb{P}_i\}$ for each $1 \leq i \leq M$ and $i \notin Q$. Finally, $\mathcal{F}_j$ forwards $(\mathcal{N}, \{(Z_i, H_i, W_i, J_i)\}_{i \notin Q}, Z_{\hat{i}}, W_{\hat{i}}, J_{\hat{i}}, \{H_i\}_{i \in Q})$ to the CS-server.

When receiving the deduplicated reports $(\mathcal{N}, \{(Z_i, H_i, W_i, J_i)\}_{i \notin Q}, Z_{\hat{i}}, W_{\hat{i}}, J_{\hat{i}}, \{H_i\}_{i \in Q})$ from $\mathcal{F}_j$, the CS-server forwards them to the customer $\mathcal{C}$.

4.4.1.6 Data-Reading

When $\mathcal{C}$ receives the deduplicated reports from the CS-server, $\mathcal{C}$ uses k to decrypt the deduplicated reports and distributes the rewards $\mathcal{T}_b$ to the contributors (mobile users) as follows:

- ▷ For each $(Z_i, W_i, J_i) \in \{(Z_i, W_i, J_i)\}_{i \notin Q}$, $\mathcal{C}$ computes

$$\mathcal{S}_i = SD(\mathcal{H}(W_i||W_i^k), J_i), \quad \mathcal{P}_i = \mathcal{SD}(\mathcal{H}(\mathcal{N}||\mathcal{S}_i)||Z_i).$$

After recovering $\mathcal{P}_i$, $\mathcal{C}$ checks whether

$$\hat{e}(H(\mathcal{P}_i), X_j) \stackrel{?}{=} \hat{e}(\mathcal{S}_i, g). \tag{4.4}$$

If yes, $\mathcal{C}$ accepts $\mathcal{P}_i$ and believes $\mathcal{U}_i$ actually generates $\mathcal{P}_i$.

▷ For $(Z_{\hat{i}}, W_{\hat{i}}, J_{\hat{i}})$, $\mathcal{C}$ computes

$$\mathcal{S}_{\hat{i}} = SD(\mathcal{H}(W_{\hat{i}}||W_{\hat{i}}^k), J_{\hat{i}}), \quad \mathcal{P}_{\hat{i}} = \mathcal{SD}(\mathcal{H}(\mathcal{N}||\mathcal{S}_{\hat{i}}), Z_{\hat{i}}).$$

After recovering $\mathcal{P}_{\hat{i}}$, $\mathcal{C}$ checks whether

$$\hat{e}(H(\mathcal{P}_{\hat{i}}), X_j) \stackrel{?}{=} \hat{e}(\mathcal{S}_{\hat{i}}, g). \tag{4.5}$$

If yes, $\mathcal{C}$ accepts $\mathcal{P}_{\hat{i}}$ and believes $\{\mathcal{U}_i\}_{i\in Q}$ actually generate $\mathcal{P}_{\hat{i}}$.

After obtaining $(\{\mathcal{P}_i\}_{i\notin Q}, \mathcal{P}_{\hat{i}})$, $\mathcal{C}$ determines the rewards that the participating mobile users can acquire based on their contributions. Suppose the mobile user with H_i can earn $\mathcal{B}_i$. $\mathcal{C}$ sends the items $\{(\mathcal{N}, H_i, \mathcal{B}_i)\}_{1\leq i\leq M}$ to the CS-server.

When a mobile user $\mathcal{U}_i$ retrieves the earned rewards, $\mathcal{U}_i$ sends $(\mathcal{N}, H_i)$ to the CS-server. The CS-server randomly picks $l_i' \in \mathbb{Z}_p$ and returns it to $\mathcal{U}_i$. After receiving l_i', $\mathcal{U}_i$ computes $b_i' = v_i^{-1}(v_i b_i + l_i - l_i')$ and returns b_i' to the CS-server. Then, the CS-server calculates $H_i' = \mathcal{G}^{l_i'} U_i^{b_i'}$, finds the item $(\mathcal{N}, H_i, \mathcal{B}_i)$, in which $H_i' = H_i$, and returns the corresponding rewards $\mathcal{B}_i$ to $\mathcal{U}_i$. In addition, if $\mathcal{U}_i$ tries to double-retrieve the benefit, which means that there is another b_i'' computed by $\mathcal{U}_i$ for a random challenge l_i'', the secret key of $\mathcal{U}_i$ is easy to be recovered by the CS-server as $v_i = \frac{l_i''-l_i'}{b_i'-b_i''}$.

4.4.2 Security Analysis

The extended Fo-SDD only exposes the knowledge that some anonymous mobile users have submitted identical sensing reports. This is the best result supporting data deduplication with high security guarantee currently. Now we discuss the security properties of the extended Fo-SDD.

Secure Data Deduplication: The method to realize data deduplication in the extended Fo-SDD remains the same as that in Fo-SDD. The fog node is able to detect the replicate reports based on the ciphertexts if the sensing data is identical. The improved security of the extended Fo-SDD is analyzed as follows:

▷ No "Duplicate-Linking" Leakage: In the extended Fo-SDD, we use the blind signature [23] to protect the identities of mobile users and thereby prevent information leakage from the equality of sensing reports. Specifically, in Service-Setup phase, the CS-server generates the anonymous credentials for mobile users using blind signatures and each mobile user utilizes the credential to prove its capability to join crowdsensing activities in Data-Collection phase without exposing its identity. To prove the unforgeability of the credential, we assume that the zero-knowledge proof $\mathcal{SPK}$ is sound, that is, there is an extract algorithm $\mathcal{EX}$ to capture the witness used by the mobile user. In the credential generation query, an adversary can obtain a valid credential on any identity with the aid of a simulator, who can use $\mathcal{EX}$ to extract the witness from the proof. If the adversary can forge a valid credential, the simulator can utilize this credential to forge a valid blind signature within a non-negligible probability. However, since the blind signature is unforgeable under the q-Strong Diffie-Hellman (q-SDH) assumption [23], it is intractable for the adversary to forge a valid credential. Therefore, the mobile users are anonymous in the extended Fo-SDD, as long as the q-SDH assumption holds. In short, even if a curious entity can learn the equality of sensing reports, it cannot link these duplicates to specific mobile users. Therefore, there is no "duplicate-linking" leakage in the extended Fo-SDD.

▷ Security against "Duplicate-Replay" Attacks: To prevent "duplicate-replay" attacks, $\mathcal{F}_j$ checks whether $V_i = V_i'$ and $Y_i = Y_i'$ in two given sensing reports $\mathbb{P}_i = (\mathcal{N}, Z_i, V_i, T_i, H_i, W_i, J_i)$ and $\mathbb{P}'_i = (\mathcal{N}, Z_i', V_i', T_i', H_i', W_i', J_i')$. If $V_i = V_i'$ and $Y_i = Y_i'$, these reports are the same and from the same mobile user, such that one of the reports may be replayed. Since a greedy mobile user does not have $\mathcal{P}_i$, the user cannot obtain $\mathcal{S}_i$. Thus, the user is unable to generate a new sensing report from a captured one. The greedy mobile user only can replay the captured one, what can be detected by $\mathcal{F}_j$. To ensure all encryption keys can correctly decrypt the sensing reports, $\mathcal{C}$ verifies the recovered data by checking $\hat{e}(H(\mathcal{P}_i), X_j) \stackrel{?}{=} \hat{e}(\mathcal{S}_i, g)$. If the equation holds, the sensing data is recovered correctly. Therefore, the extended Fo-SDD is secure against "duplicate-replay" attacks.

Efficient Contribution Claim: In the extended Fo-SDD, we allow honest anonymous mobile users to claim the contributions and retrieve the rewards from the CS-server, and prevent a greedy mobile user from double-reporting the sensing data or double-retrieving rewards. The security of contribution claim is discussed as follows:

▷ Detection of Double-Reporting: We design a double-reporting tag $T_i = \mathcal{G}^{v_i} G^{Y_i l_i}$ for each sensing report. If a greedy mobile user $\mathcal{U}_i$ submits two sensing reports in a

reporting period to the CS-server, there are two different pairs (Y_i, T_i) and (Y_i', T_i'), but the same l_i in two reports. Having (Y_i, T_i) and (Y_i', T_i'), it is easy to recover the public key of $\mathcal{U}_i$, that is, $U_i = (\frac{(T_i)^{Y_i'}}{(T_i')^{Y_i}})^{\frac{1}{Y_i'-Y_i}}$. In addition, the CS-server cannot slander an honest mobile user. To do it, a new T_i' should be computed for the CS-server. Nonetheless, it is difficult for the CS-server to compute T_i' without a valid l_i. Therefore, if the pseudo-random function F is secure, the CS-server cannot successfully slander an honest mobile user.

- ▷ Detection of Double-Retrieving: Chameleon hash function [129] is employed to enable mobile users to claim their contributions and discover greedy mobile users who double-retrieve the rewards. The Chameleon hash function is $H_i = \mathcal{G}^{l_i} U_i^{b_i}$, which is a secure hash function based on Discrete Logarithm (DL) assumption in group $\mathbb{G}_T$. To retrieve the rewards, $\mathcal{U}_i$ can use its secret key v_i to open the hash function with (b_i', l_i'). However, a greedy anonymous mobile user who double-retrieves the rewards would be traced, when the CS-server has two items (b_i', l_i') and (b_i'', l_i''), that is, $v_i = \frac{l_i''-l_i'}{b_i'-b_i''}$. Besides, it is also impossible for the CS-server to slander an honest mobile user, since it cannot generate a valid item (b_i'', l_i'') without the user's secret key v_i.

In summary, the extended Fo-SDD supports sensing data deduplication with high security guarantee, and efficient contribution claim with the detection of double-reporting mobile users or double-retrieving mobile users.

4.5 Performance Evaluation

We evaluate the computational and communication overhead of Fo-SDD and extended Fo-SDD, and show the performance of fog-assisted task allocation.

4.5.1 Computational Overhead Evaluation

To evaluate the computational overhead, we implement Fo-SDD and extended Fo-SDD on a notebook with Intel Core i5-4200U CPU and the clock rate is 2.29GHz and the memory is 4.00 GB. The notebook acts as the customer, the CS-server and a fog node. We also use a HUAWEI MT2-L01 smartphone with Kirin 910 CPU and 1250M memory to run the operations on mobile devices. The operation system is Android 4.2.2 and the toolset

Table 4.1: Run time of Fo-SDD (Unit: millisecond)

Phases	$\mathcal{C}$	CS-server	$\mathcal{F}_j$	$\mathcal{U}_i$
Task-Releasing	10.329	–	–	–
Task-Allocation	–	33.534	18.649	5.732
Data-Collection	–	–	4.847	193.459
Data-Deduplication	–	1.543	6.234	–
Data-Reading	794.624	–	–	–

Table 4.2: Run time of the Extended Fo-SDD (Unit: millisecond)

Phases	$\mathcal{C}$	CS-server	$\mathcal{F}_j$	$\mathcal{U}_i$
Task-Releasing	11.043	–	–	–
Task-Allocation	–	33.968	19.425	5.653
Data-Collection	–	–	4.275	464.649
Data-Deduplication	–	1.434	6617.835	–
Data-Reading	1435.657	–	–	–

is Android NDK r8d. We use a version 5.6.1 of MIRACL library to implement number-theoretic based methods of cryptography. The Weiling pairing is utilized to realize the bilinear pairing operation and the elliptic curve is chosen with a base field size of 512 bits. The size of the parameter p is 160 bits. 50 mobile users submit sensing reports, in which 10 reports are replicate. The running time for every entity in the Fo-SDD and the extended Fo-SDD is shown in Table 4.1 and Table 4.2, respectively. It seems that the operations in Data-Deduplication phase of the extended Fo-SDD are costly for $\mathcal{F}_j$ to deal with 50 sensing reports simultaneously. But in reality, these reports are received randomly, instead of arriving at the same time. Therefore, it is still efficient for $\mathcal{F}_j$ to respond to the mobile users in Data-Deduplication phase.

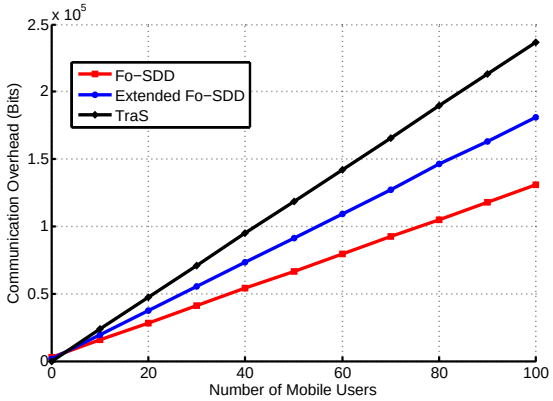
4.5.2 Communication Overhead Evaluation

We demonstrate the communication overhead of the Fo-SDD among the CS-server, $\mathcal{C}$, $\mathbb{F}$ and $\mathbb{U}$. The parameter p is set to be 160 bits. When releasing a sensing task $\mathcal{T}$, $\mathcal{C}$ sends $(C_c, K, \mathcal{T}_a)$ to the CS-server, which is $2048 + |\mathcal{T}|$ bits, where $|\mathcal{T}|$ denotes the binary length of $\mathcal{T}$. The CS-server forwards $(\mathcal{N}, C'_j, C_2, C_3, K, \mathcal{T}_a)$, whose size is $2720 + |\mathcal{T}|$ bits, to each fog node $\mathcal{F}_j \in \mathbb{F}$. After that, $\mathcal{F}_j$ sends $2208 + |\mathcal{T}|$-bit $(\mathcal{N}, D_i, K, \mathcal{T}_a)$ to each mobile user $\mathcal{U}_i \in \mathbb{U}$. After generating the sensing report, $\mathcal{U}_i$ needs to forward $1760 + |\mathcal{P}_i|$-

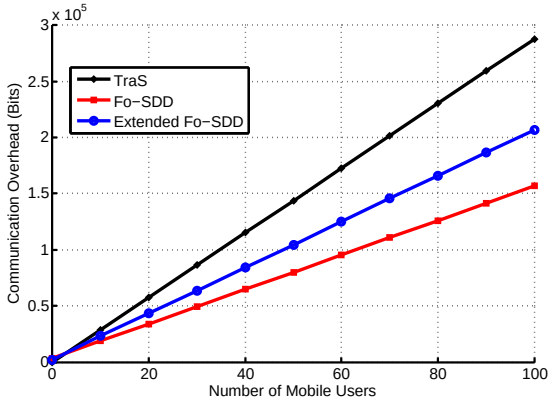
bit $(\mathcal{N}, Z_i, \sigma_i, W_i, J_i)$ to $\mathcal{F}_j$, where $|\mathcal{P}_i|$ is the binary length of $\mathcal{P}_i$. $\mathcal{F}_j$ performs the data deduplication after obtaining the reports from $\mathbb{U}$, and forwards the deduplicated reports $(\mathcal{N}, \{(\widehat{\mathbb{P}}_i, W_i, J_i)\}_{i\notin Q}, Z_{\hat{i}}, \sigma_Q, W_{\hat{i}}, J_{\hat{i}})$ to the CS server, which is of binary length $2208 + 2048(M - |Q|) + |\mathcal{P}_i|(M - |Q| + 1)$ bits, where $|Q|$ is the number of replicate reports in $\{\mathbb{P}_1, \cdots, \mathbb{P}_M\}$. Then, the CS-server sends the deduplicated reports to $\mathcal{C}$. If there is no replicate data in sensing reports, that is, $|Q| = 1$, the communication overhead between $\mathcal{F}_j$ and the CS-server is $160 + 2048M + |\mathcal{P}_i|M$ bits, as well as the burden between the CS-server and the customer $\mathcal{C}$.

The communication overhead of the extended Fo-SDD is low. The data exchanged among $\mathcal{C}$, the CS-server, $\mathcal{F}_j$ and $\mathcal{U}_i$ in Task-Releasing and Task-Allocation phases has the same length with those in the Fo-SDD. In Data-Collection phase, $\mathcal{U}_i$ needs to forward $6368 + |\mathcal{P}_i|$-bit $(\mathcal{N}, Z_i, V_i, T_i, H_i, W_i, J_i, \mathcal{SPK})$ to $\mathcal{F}_j$. $\mathcal{F}_j$ performs the data deduplication and forwards the deduplicated reports $(\mathcal{N}, \{(Z_i, H_i, W_i, J_i)\}_{i\notin Q}, Z_{\hat{i}}, W_{\hat{i}}, J_{\hat{i}}, \{H_i\}_{i\in Q})$ to the CS server, which is of binary length $1184+2048(M-|Q|)+|\mathcal{P}_i|(M-|Q|+1)+1024|Q|$ bits, where $|Q'|$ is the number of replicate and non-replayed reports in $\{\mathbb{P}_1, \cdots, \mathbb{P}_M\}$. Then, the CS-server forwards the deduplicated reports to $\mathcal{C}$.

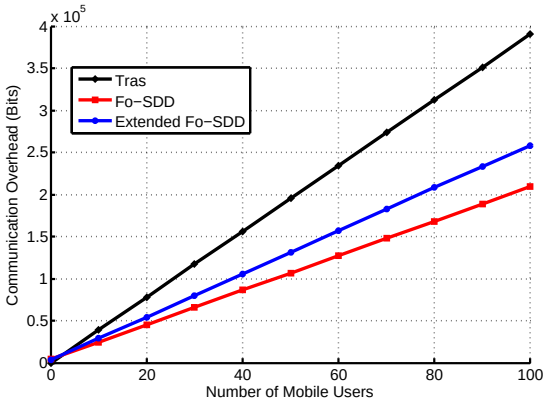
We compare Fo-SDD, extended Fo-SDD and TraS (AES [20] is used to encrypt the sensing data and DSS [20] is used to claim contributions) about the communication overhead between the CS-server and the fog nodes. Fig. 4.4 shows the comparison of the TraS, Fo-SDD and extended Fo-SDD, when 50% of sensing reports are replicate, and each mobile user delivers one sensing report to fog nodes. The length of sensing reports is 512 bits, 1024 bits and 2048 bits in Fig. 4.4(a), Fig. 4.4(b) and Fig. 4.4(c), respectively. With the increasing number of mobile users participating in crowdsensing activities, Fo-SDD and extend Fo-SDD can reduce a large number of communication overhead compared with the TraS. The Fo-SDD has the best communication efficiency as the replicate sensing data are deleted and the signatures of mobile users who report the replicate data are aggregated. Fig. 4.5 illustrates the comparison of the TraS, Fo-SDD and extended Fo-SDD about the communication overhead, when 50 mobile users submit 50 sensing reports. The length of reports is 512 bits, 1024 bits and 2048 bits in Fig. 4.5(a), Fig. 4.5(b) and Fig. 4.5(c), respectively. With the increasing percentage of replicate sensing reports, Fo-SDD and extend Fo-SDD can save plenty of communication bandwidth compared with the TraS. As shown in Fig. 4.4 and Fig. 4.5, the Fo-SDD is the most efficient scheme in three ones. Fig. 4.6 shows the relation among the TraS, Fo-SDD and extended Fo-SDD in terms of communication overhead under the various percentage of replicate data, the report length and the number of mobile users. In Fig. 4.6 (a), if the points determined by the number of mobile users and the percentage of duplicates are located in the red area, the communication overhead of Fo-SDD between the CS-server and fog nodes is lower than that of TraS;



(a) Communication Overhead with $|\mathcal{P}_i|$=512

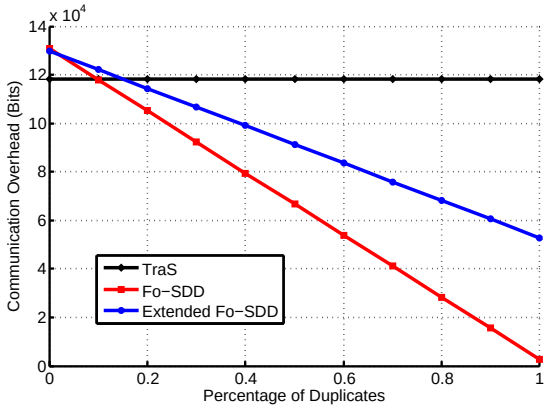


(b) Communication Overhead with $|\mathcal{P}_i|$=1024

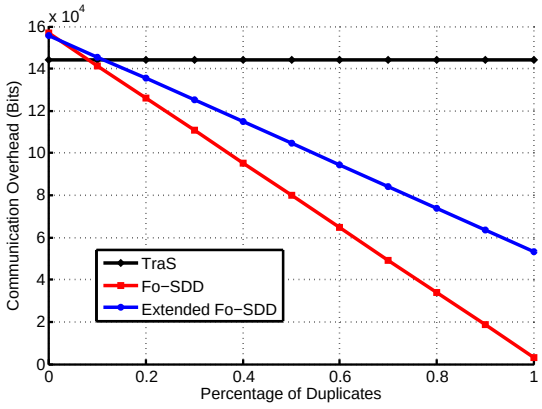


(c) Communication Overhead with $|\mathcal{P}_i|$=2048

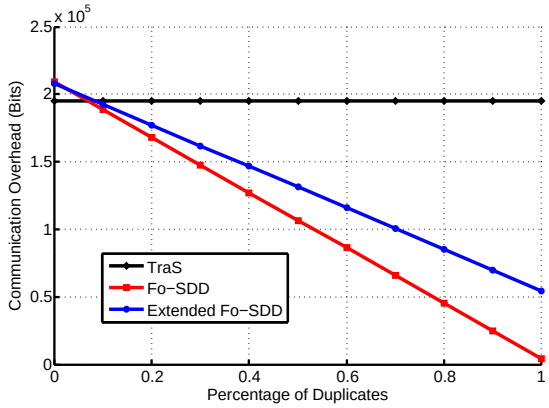
Figure 4.4: Comparison Results on Communication Overhead between Fog and CS-server with $Q/M = 50\%$



(a) Communication Overhead with $|\mathcal{P}_i|$=512

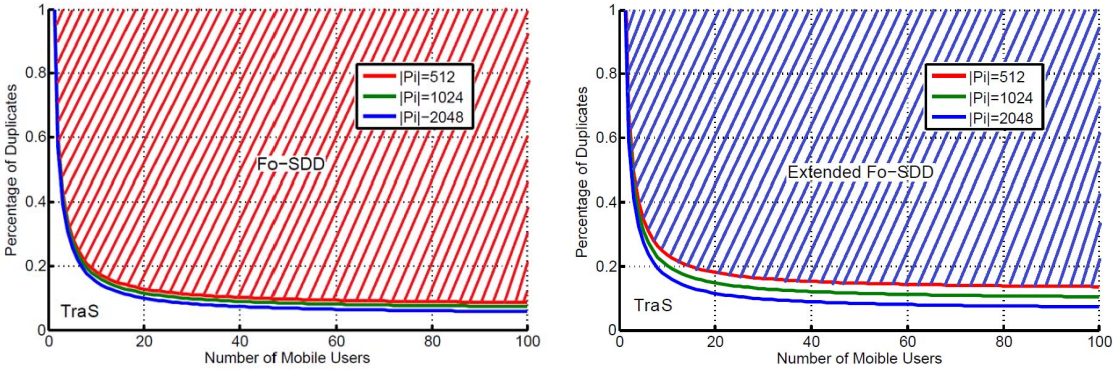


(b) Communication Overhead with $|\mathcal{P}_i|$=1024



(c) Communication Overhead with $|\mathcal{P}_i|$=2048

Figure 4.5: Comparison Results on Communication Overhead between Fog and CS-server with 50 Mobile Users



(a) Relation between TraS and Fo-SDD

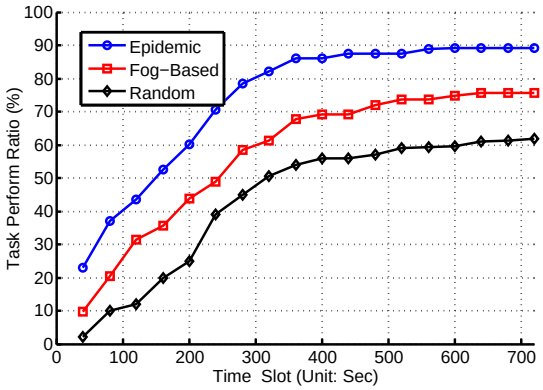
(b) Relation between TraS and Extended Fo-SDD

Figure 4.6: Relation among TraS, Fo-SDD and Extended Fo-SDD

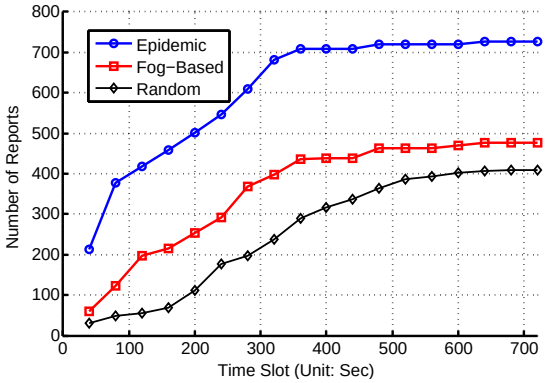
otherwise, TraS is more communication-efficient than Fo-SDD. In Fig. 4.6 (b), if the points determined by the number of mobile users and the percentage of duplicates are located in the blue area, the extended Fo-SDD is more efficient than the TraS on the communication overhead, and the TraS has lower communication overhead than the extended Fo-SDD, if the points are located in the opposite area. For example, if there are 60 mobile users to deliver 60 sensing reports and the percentage of duplicates is 60%, the Fo-SDD is more efficient than the TraS as shown in Fig 4.6(a), and the extended Fo-SDD is more efficient than the TraS as shown in Fig 4.6(b). In addition, the length of sensing reports does not have a big impact on the relations of communication efficiency for the TraS, Fo-SDD and extended Fo-SDD.

4.5.3 Performance of Task Allocation

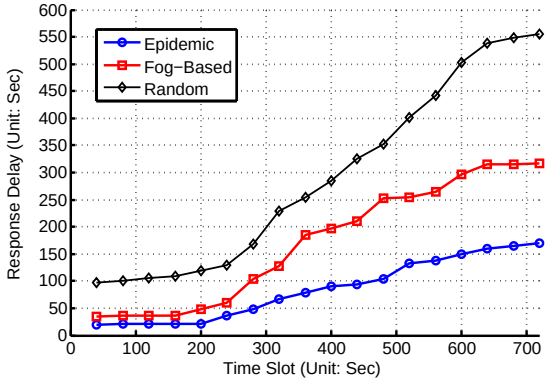
We conduct a simulation to show that the fog-assisted task allocation approach can improve the accuracy of sensing tasks assignment. The simulation is conducted on Infocom06 trace [133], which formalizes the mobility pattern of mobile users. The setting is similar with the simulation in [134]. We compare the fog-assisted task allocation approach with two methods. One is epidemic allocation, in which the CS-server allocates the tasks to all the mobile users connected with and the mobile users perform the tasks straightway; the other is random allocation, where the SC server randomly chooses 5 mobile users to perform the tasks. As shown in Fig. 4.7(a), Fig. 4.7(b) and Fig. 4.7(c), fog-assisted method has a higher task perform ratio, receives more crowdsensing reports and has lower delay



(a) Comparison on Task Perform Ratio



(b) Comparison on Number of Reports



(c) Comparison on Response Delay

Figure 4.7: Performance on Fog-Assisted Task Allocation

to accomplish the tasks than the random allocation. Although the epidemic method can get higher perform ratio and lower delay than the fog-assisted method, the CS-server may receive a large amount of reports that are collected out of the sensing area, which results in the waste of precious bandwidth and storage resources.

4.6 Summary

We have developed a fog-assisted mobile crowdsensing (Fo-MCS) framework to improve the accuracy of task allocation with the aid of fog nodes. We have also proposed a fog-assisted secure data deduplication scheme (Fo-SDD) to reduce the communication overhead between fog nodes and CS-server. The Fo-SDD enables fog nodes to detect and erase the replicate data in sensing reports, and provides high security guarantee against brute-force attacks and "duplicate-replay" attacks. To resist "duplicate-linking" leakage, we have extended the Fo-SDD to hide the identities of mobile users, such that no attacker can link the identical sensing reports to specific mobile users. In addition, we have leveraged Chameleon hash function to achieve contribution claim and reward retrieval for anonymous mobile users. Finally, we have discussed the security and efficiency of the proposed schemes and demonstrated the advantages of the Fo-MCS framework.

Chapter 5

Privacy-preserving Data Statistics

5.1 Introduction

Smart grid integrates the power grid with information and communication technologies, e.g., network communication, control systems and computation facilities, to achieve two-way electricity and information exchange between operation centers and smart meters, while making the grid more reliable, efficient, secure and greener [135]. It enables operation centers to measure, collect and analyze real-time energy consumption and local electricity generation for energy distribution management, state estimation, outage identification and dynamic billing. The operation centers expose the customers' electricity consumption to power plants, which may help them to adjust the energy production and reduce the need to fire up costly and secondary power plans. The customers not only access their real-time usage information and electricity prices, but also decrease their electricity costs by shifting the uninterrupted activities from peak time to non-peak time [136, 137].

Although power usage data collection promotes the balance between supply and demand, it brings serious privacy issues toward customers, as it is possible to infer the customers' daily activities, habits and other privacy witnessable references from power consumption data. A relatively low and static daily consumption of a household may indicate that no one is at home [138]; a conspicuous drop of power consumption at midnight may indicate the households go to sleep [139]. The determination of personal behavior patterns is a serious privacy concern in smart grid defined by Electronic Privacy Information Center. To preserve customers' behavior patterns, IEC 62351 [140] resists eavesdropping attacks using TLS encryption [141], including AES CBC, AES GCM or 3DES EDE CBC. Ontario Information Technology Standards suggest to use IPsec or TLS to provide authen-

tication, privacy protection, integrity checking and replay protection for advanced metering communications.

Traditional data encryption increases the data size of consumption reports and causes heavy communication overhead. To address this issue, privacy-preserving data aggregation schemes [142, 143, 144] have been proposed to compress the consumption reports at local collectors and forward them in a compact form to operation centers. These schemes achieve end-to-end confidentiality of meter readings, but sacrifice the integrity of consumption reports, indicating that they cannot provide sufficient integrity protection on consumption reports against misbehaving collectors. Unfortunately, the consumption reports are transmitted on public networks, such as Cellular and the Internet, with the storage and forwarding of collectors, according to Toronto Hydro. The collectors are vulnerable to be hacked by attackers. Malicious attackers can inject false data into the aggregated reports or corrupt the meter readings without being detected, and thereby affect state estimation, break power dispatch and control electricity prices through misbehaving collectors [110]. The power outage in Ukraine on Dec. 23, 2015 caused by a devastating cyber attack on a power station warns us that any vulnerability in advanced metering infrastructure may be exploited by hackers to create a blackout. Misbehaving collectors have not being paid enough attentions lately. A handful of schemes [145, 146] aimed to reduce the dependence on a single collector, but bring heavy communication overhead to distribute the reliability to multiple collectors by means of secret sharing. Further, once the consumption reports of different customers are aggregated, it is impossible to achieve dynamic billing for customers. Therefore, it is of importance to design an efficient smart metering scheme that simultaneously supports data aggregation and dynamic billing with high security guarantee.

To balance security and efficiency, we propose a Privacy-Preserving Smart Metering scheme (P^2SM) [109] to achieve end-to-end security, data aggregation and dynamic billing, simultaneously. Considering a realistic case that the collectors deployed at public areas may be controlled by attackers, we build a new security model between traditional semi-honest model and malicious model to formally define the misbehavior of collectors. We achieve the authentication, confidentiality and integrity of consumption reports against misbehaving collectors for smart metering based on Chameleon hash function [147], proxy re-encryption [22] and homomorphic authenticators [148]. In addition, we upgrade the collectors with computing and storage resources, such that they can temporarily maintain individual reports for dynamic billing. Our contributions can be summarized as four folds:

- ▷ Inspired by the fact that the collectors at public areas may be hacked, we introduce a stronger security model to formalize the collectors' misbehavior in reality. Different

from semi-honest adversaries, the misbehaving collectors are not only interested in the personal behavior patterns of customers, but also launch pollution attacks to insert false data into normal meter readings to corrupt state estimation of operation centers.

- ▷ To prevent pollution attacks from collectors, we design the P^{2}SM by leveraging proxy re-encryption [22] and homomorphic authenticators [148]. The privacy-preserving data aggregation is achieved to prevent privacy leakage and reduce communication overhead. P^{2}SM does not allow the collectors to generate their signatures by themselves, but aggregate the smart meters' signatures to guarantee the integrity of the aggregated consumption reports. As a result, a misbehaving collector cannot inject false data into the consumption reports or invade the privacy of customers.

- ▷ Once smart meters' signatures are aggregated, it is impossible to offer message authentication for customers. We design an identity authentication mechanism from Chameleon hash function [147] for smart metering. With the desirable property of homomorphism, the authentication messages of different customers can be aggregated to further improve communication efficiency.

- ▷ To support dynamic billing, P^{2}SM enables collectors to use the maintained individual consumption reports to generate verifiable daily bills for customers. Specifically, the collectors aggregate the consumption reports of each customer with the electricity prices to generate daily bills, and submit them to the operation center. The operation center transforms the encrypted bills to be readable for customers. Furthermore, the customers can verify the correctness of their daily bills to detect the corruption of misbehaving collectors and greedy utilities.

5.2 Problem Statement

We formalize system model, present security threats and identify design goals.

5.2.1 System Model

We formalize power consumption data collection for operation centers and dynamic billing for energy companies as depicted in Fig. 5.1. Energy companies (utilities) have a good supply of electricity from plants and provide electricity retailing services to customers.

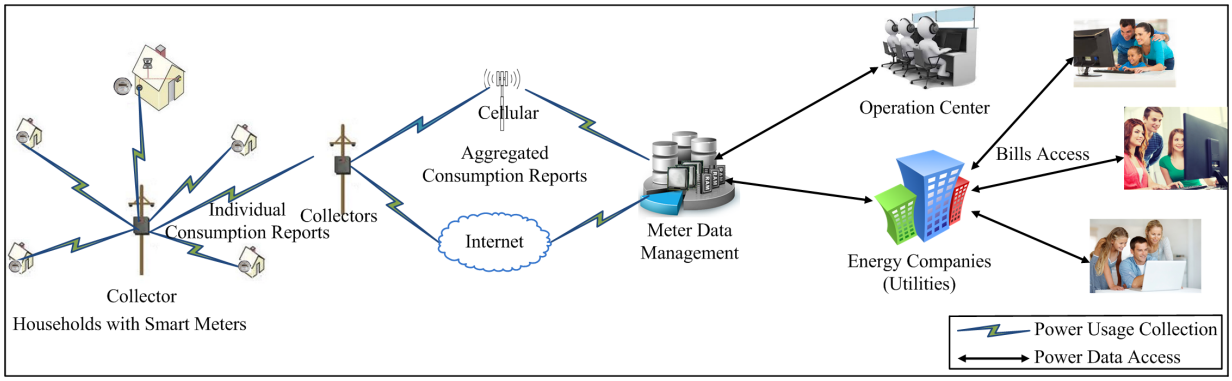


Figure 5.1: System Model for Smart Metering

To realize real-time power dispatch, operation centers collect, analyze and process real-time power usage data of customers and monitor the power consumption through varying electricity prices. A temper-proof smart meter is installed in each customer's house to measure real-time power consumption and submit the meter readings to operation centers every ρ minutes, in general, $\rho = 15$ or 60. A local collector, which is a wireless access point or base station, is deployed to connect the operation center and smart meters in a home area network. In each reporting time slot, smart meters measure meter readings and deliver consumption reports to the collector through relatively inexpensive WiFi or ZigBee technologies. After receiving the individual consumption reports, the collector transiently stores them, aggregates these reports into a compact one, and delivers the aggregated report to the operation center several times a day through wired network, e.g., the Internet. According to these aggregated reports, the operation center monitors electricity distribution and determines dynamic electricity prices. The electricity price is returned to the collector per day, and the collector computes the daily bills of customers using the maintained individual consumption and sends the electricity bills to the utility. Finally, the customers access their electricity bills via the Internet and regulate their daily activities to decrease electricity costs.

5.2.2 Security Model

As the intermediates in advanced metering infrastructure, local collectors are deployed at the public areas and they are vulnerable to be hacked by malicious hackers. The hackers may invade customers' privacy, inject false data, corrupt state estimation and control electricity prices through the misbehaving collectors. To be more close to the reality, we

define a new security model with a misbehaving adversary that has rational attack behaviors. The misbehaving collector may be neither completely malicious, to block the power usage data transmission that can be quickly detected by the operation center, nor just honest-but-curious, to be interested in customers' living patterns through eavesdropping. The hacked collector is more powerful than the honest-but-curious adversary, and more rational than the malicious adversary. On one hand, to prevent its misbehavior being identified, a misbehaving collector will follow the communication protocols and pretend to be honest; on the other hand, it tries to use all sorts of methods to achieve the goals of ulterior motives. Specifically, a misbehaving collector launches the following attacks to invade customers' privacy and corrupt state estimation in smart grid:

- ▷ A misbehaving collector learns the customers' privacy via eavesdropping.
- ▷ A misbehaving collector injects false data into power consumption in home area network to corrupt state estimation or control electricity prices.
- ▷ A misbehaving collector may forge the smart meters' individual reports or aggregated reports to corrupt state estimation of the operation center.
- ▷ A misbehaving collector may forge the daily electricity bills to cheat the operation center, utilities and customers.

The eavesdropping attack and forgery attack have been discussed in the existing literatures [149]. The pollution attack is brightly new in our security model. Therefore, we utilize the following game between the misbehaving adversary and the advanced metering infrastructure to formally define this attack:

1. The advanced metering infrastructure setups the whole system to collect the power consumption of customers in a home area network.
2. The adversary can interact with the system and query the individual consumption reports, providing, for each query, a smart meter and its reading. The system generates the individual report for each query and returns the report to the adversary.
3. Finally, the adversary outputs an aggregated report different from the aggregation of queried individual consumption reports.

If the adversary is able to generate a valid aggregated report that is not the aggregation of queried individual reports with non-negligible probability, we say that the adversary

wins the game. The smart metering scheme is able to resist pollution attacks if for any misbehaving collector the probability that the collector wins the above game is negligible.

The smart meters are physically protected to prevent customers from stealing electricity. The malfunction of smart meters would be discovered and replaced by utilities in time. In addition, the customers are honest to purchase the electricity from utilities and access their daily electricity bills. The operation center, fully controlled by the government, is honest to manage power transmission and balance the electricity demand and response. The damage of the operation center may directly impact national security and social stability, thereby sufficient security policies are implemented to protect the operation center. The utilities are honest to provide electricity retailing services to customers, while they are curious on customers' privacy and greedy on their benefits, such as increasing their income by modifying customers' bills.

5.2.3 Design Goals

To enable privacy-preserving smart metering under the aforementioned system model and resist various security threats, P^{2}SM should achieve the following objectives:

- ▷ *Authentication*: To assure that individual consumption reports are from legal customers. It is impossible for an attacker to deliver a forged consumption report acceptable for the operation center.
- ▷ *Privacy Preservation*: To guarantee that no attacker is able to learn the meter readings and thereby invade the privacy of customers, even it either eavesdrops on communication channels or hacks the collectors. The curious utility cannot learn the power consumption of their customers, except the daily electricity bills.
- ▷ *Integrity Checking*: To ensure that neither individual consumption reports nor aggregated reports can be modified by attackers. Even the misbehaving collectors are not able to corrupt the integrity of consumption reports by injecting false data into normal meter measurements. Therefore, the operation center can obtain correct power consumption data.
- ▷ *Dynamic Billing*: To achieve that the daily bills are generated from individual consumption reports and fluctuant electricity prices. The customers can access their bills and verify the correctness.

▷ *Efficiency*: The communication cost is required to be low to save energy during data transmission and guarantee that the operation center receives the consumption reports within short delay. In addition, there should be no time-consuming operation for smart meters due to their constrained computational capabilities.

5.3 P^{2}SM

We describe a high-level description of P^{2}SM to briefly show the work flow and information flow, and then give the detailed description of P^{2}SM.

5.3.1 High-level Description

The reason that the existing privacy-preserving data aggregation schemes in smart grid are vulnerable to the pollution attack is that the collector generates the signature on the aggregated consumption report by itself to ensure the report integrity. If the collector becomes dishonest, it can arbitrarily insert forged data into the aggregated report without being detected by the operation center. To prevent this attack, we extend the homomorphic authenticators in [148] to be pairing-based cryptosystem to achieve the aggregation of the signatures of various measurements generated by different smart meters, which is a big challenge if no parameter is pre-shared among smart meters [150]. To overcome this challenge, we first allow the smart meters to sign the meter measurements rather than their ciphertexts using the individual secret keys based on the homomorphic authenticators [148], and then enable the collectors to re-sign the individual signatures to generate signatures under a common key selected by the operation center for the smart meters in the home area network based on bilinear pairing. Thereby, the re-signed signatures can be aggregated to prevent the misbehaving collector from corrupting the meter measurements. Unfortunately, after the individual signatures are re-signed and aggregated, they cannot offer the authentication for smart meters. To fix this drawback, we design a new identity authentication mechanism based on the Chameleon hash function [147] with batch verification, resulting in the reduction of computational and communication overhead.

In addition, it is of difficulty for the operation center to generate daily bills after the individual consumption reports are compressed. To resolve this problem, we novelly upgrade the capability of collectors with storage spaces. Hence, these individual reports transiently maintained on the collectors can be used to compute the daily bills with the fluctuant electricity prices. To delegate the decryption of daily bills, the proxy re-encryption [22] is

leveraged to enable the operation center to re-encrypt the bills generated by the collectors for the customers on behalf of a proxy. With the homomorphism of the proxy re-encryption [22], the ciphertexts of meter readings can be aggregated to improve the communication efficiency. Moreover, to prevent the misbehaving collector from generating cheating bills, the individual signatures of meter measurements are aggregated with the prices to generate verifiable tags on the daily bills. Therefore, the customers can check whether the daily bills are correctly computed and identify the corrupted ones.

P^{2}SM consists of six phases, namely, System Initialization, Customer Registration, Report Generation, Report Aggregation, Report Reading and Dynamic Billing. The information flow of P^{2}SM is depicted in Fig. 5.2.

- ▷ System Initialization: The operation center bootstraps the whole system for smart metering and generates the public parameters $Params$ and its secret-public key pair (k, K).
- ▷ Customer Registration: The customer C_i with an installed smart meter SM_i on the house registers at the operation center using the registration message $(SM_i, y_i, \mathcal{H}_i, z_{i1}, z_{i2})$, in which $\mathcal{H}_i$ is the commitment generated from the Chameleon hash function and (z_{i1}, z_{i2}) is the ciphtetext of a random key k. The operation center returns (SM_i, RK_i) to the local collector, where RK_i is the re-sign key used to transform C_i's signature to a signature under the common key α selected by the operation center for the smart meters in the home area network.
- ▷ Report Generation: The smart meter SM_i reads the measurement m_{it} at a time slot t and generates a consumption report $P_{it} = \mathcal{U}||SM_i||a'_{it}||c_{it}||e_{it}||\sigma_{it}||t$, in which a'_{it} is the authentication message, (c_{it}, e_{it}) is the ciphertext of m_{it} and σ_{it} is the signature on m_{it}. SM_i sends P_{it} to the local collector.
- ▷ Report Aggregation: The collector aggregates the authentication messages, ciphertexts and signatures in the individual consumption reports during each forwarding period Q to generate an aggregated report $P = \mathcal{C}||a||c||e||\sigma||Q$ using the re-sign keys RK_i of all smart meters in its home area network, and forwards P to the operation center.
- ▷ Report Reading: The operation center checks the validity of the aggregated authentication message a, decrypts the aggregated ciphertext (c, e) and verifies the aggregated signature σ. Finally, the operation center obtains the total power consumption m in the home area network for state estimation and demand response.

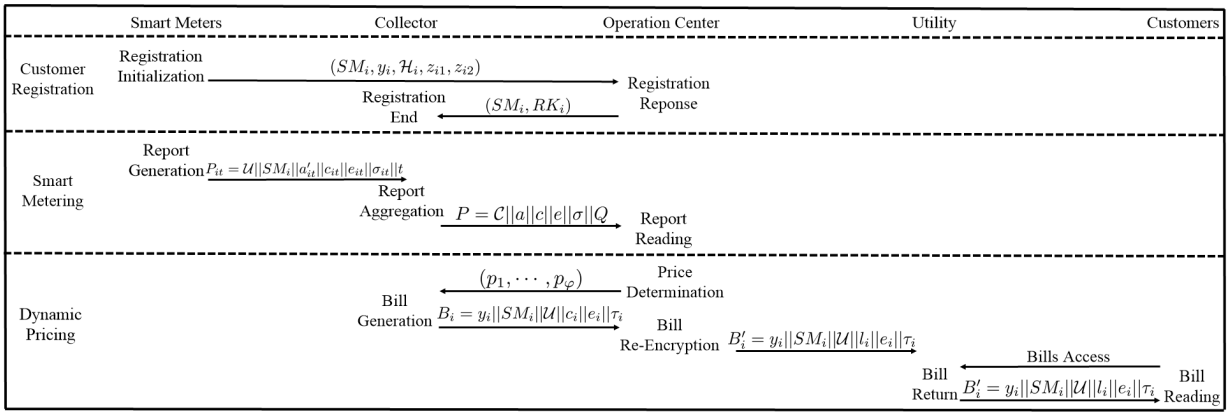


Figure 5.2: Information Flow of P^{2}SM

▷ Dynamic Billing: The operation center determines the fluctuant electricity prices $(p_1, \cdots, p_\varphi)$ during a day. The collector aggregates the ciphertexts of meter readings with the prices to generate the daily bill (c_i, e_i) for the customer, and aggregates the signatures with the prices to obtain a verifiable tag τ_i on the daily bill. To enable the customer to read the bill, the operation center re-encrypts the daily bill $B_i = y_i||SM_i||\mathcal{U}||c_i||e_i||\tau_i$ to generate $B'_i = y_i||SM_i||\mathcal{U}||l_i||e_i||\tau_i$ for the customer. Therefore, the customer can read the daily bill and uses τ_i to check the correctness of the daily bill.

5.3.2 The Detailed P^{2}SM

We then describe the construction of P^{2}SM in detail.

5.3.2.1 System Initialization

The operation center (OC) provides electricity distribution and demand response for the customers $\mathbb{C} = \{C_1, \cdots, C_N\}$ in the residential area $\mathbb{RA}$. Suppose that these customers buy electricity from a utility $\mathcal{U}$, (it is compatible that $\mathbb{C}$ use the power offered by multiple utilities). OC bootstraps the advanced metering infrastructure on behalf of a trust authority. Concretely, OC first determines the security parameter κ, which denotes the security level of the system and κ is 160 or 256 usually. OC chooses a large prime p, where $|p| = \kappa$. OC also generates two cyclic groups $(\mathbb{G}, \mathbb{G}_T)$ with the same order p. g, g_0 are two generators

of $\mathbb{G}$, and $\hat{e} : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ is a bilinear map. $H : \{0,1\}^* \to \mathbb{G}$ and $H_1 : \{0,1\}^* \to \{0,1\}^\kappa$ are cryptographic hash functions and $F : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p$ is a pseudo-random function. (E_s, D_s) are the encryption and decryption algorithms of AES cryptosystem. Then, OC randomly chooses $k \in \mathbb{Z}_p$ to compute $K = g^k$. Finally, OC releases the public parameters:

$$Params=\{p, \mathbb{G}, \mathbb{G}_T, \hat{e}, g, g_0, H, H_1, F, E_s, D_s, K\},$$

and keeps the secret key k in private.

5.3.2.2 Customer Registration

When a customer $C_i \in \mathbb{C}$'s house in the $\mathbb{RA}$ connects the smart grid, OC installs a smart meter SM_i for C_i. In the registration, C_i first randomly chooses $x_i \in \mathbb{Z}_p$ as the private key and computes the corresponding public key as $y_i = g^{x_i} \in \mathbb{G}$. The secret key x_i is plugged into SM_i or stored in a trusted platform module (TPM) integrated into the smart meter SM_i and the public key certificate is publicly accessed by any entity. Then, SM_i randomly picks $a_i, b_i \in \mathbb{Z}_p$ to compute a Chameleon hash value as $\mathcal{H}_i = g^{a_i} y_i^{b_i}$. After that, SM_i chooses two random values $k_i, r_i \in \mathbb{Z}_p$ to calculate $z_{i1} = g^{r_i}$, $r'_i = H_1(z_{i1}||K^{r_i})$ and $z_{i2} = E_s(r'_i, k_i)$. Finally, SM_i sends $(SM_i, y_i, \mathcal{H}_i, z_{i1}, z_{i2})$ to OC, and keeps (a_i, b_i, k_i) in the TPM, along with x_i.

Upon receiving $(SM_i, y_i, \mathcal{H}_i, z_{i1}, z_{i2})$, OC decrypts (z_{i1}, z_{i2}) to obtain the tag k_i as $r'_i = H_1(z_{i1}||z_{i1}^k)$ and $k_i = D_s(r'_i, z_{i2})$. Then, OC randomly picks $\alpha \in \mathbb{Z}_p$ as a unique identifier of $\mathbb{RA}$ to compute a re-sign key $RK_i = y_i^\alpha$, if C_i is the first customer in $\mathbb{RA}$; otherwise, U_i uses the existing α to compute RK_i. Finally, OC sends (SM_i, RK_i) to the local collector in $\mathbb{RA}$ through a secure channel, and stores $(SM_i, y_i, \mathcal{H}_i)$ in its database and keeps $(T_i, RK_i, \alpha, g^\alpha)$ secretly.

5.3.2.3 Report Generation

To achieve real-time power dispatch, smart meters measure power consumption and deliver electricity consumption reports every ρ minutes, i.e., $\rho = 15$ or 60 ($\rho = 60$ for Toronto Hydro). Suppose that a smart meter SM_i measures the meter reading m_{it} at a time slot t. SM_i generates an individual consumption report as follows:

- ▷ Use $b'_{it} = F(H(SM_i||k_i), t)$ to compute the authentication message $a'_{it} = x_i \cdot (b_i - b'_{it}) + a_i \mod p$;

▷ Randomly pick $s_{it} \in \mathbb{Z}_p$ to generate the ciphertext of meter reading m_{it} as:

$$c_{it} = K^{s_{it}}, \quad e_{it} = \hat{e}(g_0^{m_{it}} g^{s_{it}}, g);$$

▷ Use x_i to generate a signature on the meter reading as:

$$\sigma_{it} = (H(SM_i||\mathcal{U}||t) g_0^{a'_{it}} g^{m_{it}})^{\frac{1}{x_i}}; \tag{5.1}$$

▷ Send the individual consumption report $P_{it} = \mathcal{U}||SM_i||a'_{it}||c_{it}||e_{it}||\sigma_{it}||t$ to the collector in this area.

5.3.2.4 Report Aggregation

The collector transiently maintains the received individual consumption reports from smart meters. It is required to forward the consumption reports to OC φ times per day ($\varphi = 5$ or 24). In each forwarding period Q, the collector aggregates the individual consumption reports received in Q from N smart meters in $\mathbb{RA}$ into an aggregated report P as follows:

$$c = \prod_{t \in Q} \prod_{i=1}^{N} c_{it}; \qquad e = \prod_{t \in Q} \prod_{i=1}^{N} e_{it}; \tag{5.2}$$

$$a = \sum_{t \in Q} \sum_{i=1}^{N} a'_{it} \mod p; \qquad \sigma = \prod_{t \in Q} \prod_{i=1}^{N} \hat{e}(\sigma_{it}, RK_i). \tag{5.3}$$

The collector sets $P = \mathcal{C}||a||c||e||\sigma||Q$ and forwards P to OC, where $\mathcal{C}$ is the identifier of the collector.

5.3.2.5 Report Reading

After receiving $P = \mathcal{C}||a||c||e||\sigma||Q$, OC performs the following steps to read the aggregated report P:

▷ Use each customer's unique tag k_i to compute $b^*_{it} = F(H(SM_i||k_i), t)$ and verify whether all reports are released by legitimate smart meters by checking the equation (5.4):

$$\prod_{t \in Q} \prod_{i=1}^{N} \mathcal{H}_i \stackrel{?}{=} g^a \cdot \prod_{t \in Q} \prod_{i=1}^{N} y_i^{b^*_{it}}. \tag{5.4}$$

If the equation (5.4) holds, continue to decrypt (c, e); otherwise, retrieve the individual consumption reports from the collector to find the invalid reports.

- ▷ Decrypt the aggregated ciphertext (c, e) as $M = e\hat{e}(c, g)^{-\frac{1}{k}}$ and recover the discrete log of M base $\hat{e}(g_0, g)$ using Pollard's lambda method [151] to obtain $m = \sum_{t\in Q}\sum_{i=1}^{N} m_{it}$.
- ▷ Verify whether the equation (5.5) is valid or not:

$$\sigma \stackrel{?}{=} \hat{e}(\prod_{t\in Q}\prod_{i=1}^{N} H(SM_i||\mathcal{U}||t)g_0^a g^m, g^\alpha). \tag{5.5}$$

If yes, accept m, which is the total power consumption of customers in $\mathbb{RA}$ in the period Q; otherwise, reject m and retrieve the individual consumption reports from the collector to find the corrupted reports utilizing a recursive divide-and-conquer approach.

5.3.2.6 Dynamic Billing

According to the power consumption of customers in $\mathbb{RA}$, OC determines the electricity price in every forwarding period during a day, that is, $(p_1, \cdots, p_\varphi)$, where p_j denotes the electricity price in the jth forwarding period Q_j, and sends $(p_1, \cdots, p_\varphi)$ to the collector. The collector aggregates the individual consumption reports of a customer with the electricity prices to generate an electricity bill for the customer. Specifically, for a customer C_i, the collector computes

$$c_i = \prod_{j=1}^{\varphi}\prod_{t\in Q_j} c_{it}^{p_j},\ e_i = \prod_{j=1}^{\varphi}\prod_{t\in Q_j} e_{it}^{p_j},\ \tau_i = \prod_{j=1}^{\varphi}\prod_{t\in Q_j} \sigma_i^{p_j}, \tag{5.6}$$

where $t \in Q_j$ means that the time slot t is in the reporting period Q_j, and delivers the bill $B_i = y_i||SM_i||\mathcal{U}||c_i||e_i||\tau_i$ to OC. Then, OC verifies the correctness of the bills in $\mathbb{RA}$ by verifying the equation (7):

$$\hat{e}(g_0, g)^{\sum_{j=1}^{\varphi} m_j p_j} = \prod_{i=1}^{N} e_i \hat{e}(c_i, g)^{-\frac{1}{k}}, \tag{5.7}$$

where m_j is the total power consumption of customers in $\mathbb{RA}$ in the period Q_j. If it holds, OC further computes $l_i = \hat{e}(c_i, y_i)^{\frac{1}{k}}$ and sends the bill $B_i' = y_i||SM_i||\mathcal{U}||l_i||e_i||\tau_i$ to $\mathcal{U}$. In

addition, OC can delegate $\mathcal{U}$ to perform proxy re-encryption to transform the ciphertexts of OC to be decryptable for C_i on behalf of a proxy. Specifically, OC sends $USK_i = y_i^{\frac{1}{k}}$ to $\mathcal{U}$ to enable it to compute $l_i = \hat{e}(c_i, USK_i)$ for C_i. Finally, C_i decrypts (l_i, e_i) by computing $D_i = e_i l_i^{-\frac{1}{x_i}}$ and recovering the discrete log of D_i base $\hat{e}(g_0, g)$ using Pollard's lambda method [151] to obtain $d_i = \sum_{j=1}^{\varphi} \sum_{t \in Q_j} m_{it} p_j$. To verify the correctness of d_i, C_i checks the equation (5.8):

$$\hat{e}(\tau_i, y_i) \stackrel{?}{=} \hat{e}(\prod_{j=1}^{\varphi} \prod_{t \in Q_j} H(SM_i||\mathcal{U}||t)^{p_j} g_0^{\sum_{j=1}^{\varphi} \sum_{t \in Q_j} a'_{it} p_j} g^{d_i}, g), \tag{5.8}$$

where $a'_{it} = x_i(b_i - F(H(SM_i||k_i), t) + a_i) \mod p$. If the equation (8) holds, C_i accepts the bill d_i; otherwise, rejects it.

5.4 Security Analysis

We analyze the security properties of P^{2}SM, following the security goals described in section 5.2.3, including authentication, confidentiality and integrity.

• *Authentication*: In customer registration, SM_i utilizes the ElGamal encryption to send k_i to OC. Since the ElGamal encryption is semantically secure against chosen plaintext attacks [152] based on Hash-Diffie-Hellman problem, only OC is able to recover k_i. Hence, k_i is shared between SM_i and OC. To achieve efficient authentication, the Chameleon hash function is leveraged to design the interactions between SM_i and OC. Firstly, $\mathcal{H}_i = g^{a_i} y_i^{b_i}$ is one-way, indicating that it is easy to compute $\mathcal{H}_i$ from (a_i, b_i), but no one can recover (a_i, b_i) from $\mathcal{H}_i$, as long as the Discrete Logarithm (DL) assumption [147] holds. In addition, no one is able to find a collision (a'_i, b'_i) of (a_i, b_i) to make $\mathcal{H}_i = g^{a'_i} h_i^{b'_i}$ hold without x_i in polynomial time with non-negligible probability, unless the DL problem is tractable. However, having x_i, SM_i can compute a_i from any given b_i. Therefore, if the ElGamal encryption is semantically secure and the DL problem is intractable, it is impossible for an adversary to pretend a legal smart meter to generate consumption reports without being detected by OC.

• *Confidentiality*: To prevent the customers' power consumption from being revealed, we adopt to the proxy re-encryption [22] to encrypt meter readings. Since the proxy re-encryption is proved secure against chosen plaintext attacks, the confidentiality of m_{it} is satisfied to prevent attackers from invading C_i's privacy, even attackers can eavesdrop on

communication channels and capture the ciphertexts. When the collector obtains all individual consumption reports from smart meters in $\mathbb{RA}$, it cannot recover the meter readings but aggregating the ciphertexts based on additive homomorphism to reduce communication overhead. As for OC, it can recover the sum of power consumption in $\mathbb{RA}$ by using its secret key. In Dynamic Billing phase, the collector aggregates the individual consumption reports with the electricity prices to generate the bills and forwards the results to OC or U to allow them to re-encrypt the bills to be decryptable for the customers. As the proxy re-encryption is secure based on the Computational Bilinear Inverse Diffie-Hellman (BIDH) assumption [22], the meter readings and the electricity bills are confidential against eavesdroppers and curious entities.

• *Integrity*: To resist pollution attacks, P^{2}SM should ensure the integrity of consumption reports from smart meters to the operation center. The signatures of smart meters are used to ensure the integrity of individual consumption reports, and the aggregated signature is generated from the signatures of smart meters by the collector to prevent data corruption during the transmission from the collector to the operation center. Thus, the integrity of reports depends on the unforgeability of both the individual signatures and the aggregated signature. We prove the unforgeability of the individual signatures and the aggregated signature separately.

To ensure the integrity of the individual consumption report, SM_i generates a signature using its private key as $\sigma_{it} = (H(SM_i||\mathcal{U}||t)g_0^{a'_{it}}g^{m_{it}})^{\frac{1}{x_i}}$. The unforgeability of this signature can be reduced to the Diffie-Hellman Inversion (DHI) assumption [153], that is, within non-negligible advantage, there is no probabilistic polynomial-time algorithm to solve DHI problem: given $h, h^s \in \mathbb{G}$, where $s \in \mathbb{Z}_p$, to compute $h^{\frac{1}{s}} \in \mathbb{G}$.

Theorem 1: The signature in the individual consumption report is existentially unforgeable against adaptive chosen message attacks under the security model [127], provided that the DHI problem is difficult to be addressed with a non-negligible probability in probabilistic polynomial time.

Proof. Suppose that an adversary $\mathcal{A}$ can break the existential unforgeability of the signature with a non-negligible probability, then we can construct an algorithm $\mathcal{B}$ to solve the DHI problem. Let h be a generator of $\mathbb{G}$. $\mathcal{B}$ is given $h, h^s \in \mathbb{G}$, where $s \in \mathbb{Z}_p$, its goal is to compute $h^{\frac{1}{s}}$. $\mathcal{B}$ simulates a challenger and interacts with $\mathcal{A}$ in the following way.

▷ In setup, $\mathcal{B}$ sets the public key v to $h^{\frac{s}{r}}$ and the parameters g to h^{sr_1}, g_0 to h^{sr_2}, where r, r_1, r_2 are random values chosen from $\mathbb{Z}_p$, and forwards them to $\mathcal{A}$.

▷ $\mathcal{B}$ programs a random oracle to answer hash queries. To ensure the consistency, it maintains a list of tuples to keep the queries and corresponding responses. When

receiving queries $(SM_i, \mathcal{U}, t)$ from $\mathcal{A}$, $\mathcal{B}$ flips a bias coin $\theta_i \in \{0, 1\}$, such that $\Pr[\theta_i = 0] = 1/(q_s + 1)$, where q_s is the maximum of signing queries that $\mathcal{A}$ can make. If $\theta_i = 0$, $\mathcal{B}$ computes $w_i = h^{\beta_i}$; otherwise, $\theta_i = 1$ and $\mathcal{B}$ computes $w_i = h^{s\beta_i}$, where β_i is a random value chosen from $\mathbb{Z}_p$. At last, $\mathcal{B}$ adds a tuple $(SM_i, \mathcal{U}, t, \theta_i, \beta_i, w_i)$ to the list, and returns w_i to $\mathcal{A}$.

- ▷ $\mathcal{B}$ also programs a signing oracle and maintains a list of tuples to keep the queries and responses. When $\mathcal{A}$ queries $(SM_i, \mathcal{U}, t, a'_{it}, m_{it})$, $\mathcal{B}$ firstly checks the list in hash queries. If $(SM_i, \mathcal{U}, t)$ has not been queried, $\mathcal{B}$ generates the corresponding (θ_i, β_i, w_i) for $(SM_i, \mathcal{U}, t)$. If $\theta_i = 0$, $\mathcal{B}$ aborts and returns failure; If $\theta_i = 1$, $\mathcal{B}$ sets $\sigma_{it} = h^{\beta_i r + r_1 a'_{it} r + r_2 m_{it} r}$. Observe that σ_{it} is a valid signature on $(SM_i, \mathcal{U}, t, a'_{it}, m_{it})$ under the public key $h^{\frac{s}{r}}$. Finally, $\mathcal{B}$ returns σ_{it} and adds $(SM_i, \mathcal{U}, t, \theta_i, \beta_i, a'_{it}, m_{it}, \sigma_{it})$ to the list.

- ▷ Eventually, $\mathcal{A}$ produces a message-signature pair $(SM_i, \mathcal{U}, \hat{t}, \hat{a}'_i, \hat{m}_i, \hat{\sigma}_i)$, such that no signature query has been made for $(SM_i, \mathcal{U}, \hat{t}, \hat{a}'_i, \hat{m}_i)$. If there is no tuple in hash list, $\mathcal{B}$ issues $(SM_i, \mathcal{U}, \hat{t})$ to hash query. $\mathcal{B}$ aborts and returns failure, if $\hat{\sigma}_i$ is invalid. Next, $\mathcal{B}$ finds the tuple on hash list. If $\hat{\theta}_i = 1$, $\mathcal{B}$ aborts and returns failure; otherwise, $\hat{\theta}_i = 0$ and therefore $H(SM_i || \mathcal{U} || \hat{t}) = h^{\hat{\beta}_i}$. Hence, $\hat{\sigma}_i = h^{\frac{\hat{\beta}_i r}{s}} h^{r_1 \hat{a}'_i r + r_2 \hat{m}_i r}$. Then, $\mathcal{B}$ outputs the required $h^{\frac{1}{s}} = (\hat{\sigma}_i h^{-(r_1 \hat{a}'_i r + r_2 \hat{m}_i r)})^{\frac{1}{\hat{\beta}_i r}}$

Therefore, if the DHI problem cannot be solved with a non-negligible probability in probabilistic polynomial time, no adversary can forge the signatures on individual reports. ■

The integrity of bills can be reduced to the DHI assumption as the signatures on bills are the aggregation of smart meter's signatures. If the single signature σ_{it} is unforgeable, its aggregated signature τ_i cannot be forged by attackers as well.

Then, we show that it is impossible to forge a valid aggregated report-signature pair (σ, m) in probabilistic polynomial time under the assumption of Conference-Key Sharing (CONF) [21] in group $\mathbb{G}_T$, that is, there is no probabilistic polynomial-time algorithm that solves CONF problem [21] within a non-negligible probability: given $g, g^s g^{sv} \in \mathbb{G}$, where $s, v \in \mathbb{Z}_p$, to compute $\hat{e}(g, g)^v \in \mathbb{G}_T$.

Theorem 2: The probability of generating a valid aggregated signature σ, which is not equal to the aggregation of smart meters' signatures, in probabilistic polynomial time is negligible, provided that the CONF problem is hard.

Proof. If there is a probabilistic polynomial-time adversary $\mathcal{A}$ can break the unforgeability of the aggregated signature within a non-negligible probability, we can construct an algorithm $\mathcal{B}$ to solve the CONF problem.

Let g be a generator of $\mathbb{G}$. $\mathcal{B}$ is given $g, D = g^s, D_1 = g^{sv} \in \mathbb{G}$, where $s, v \in \mathbb{Z}_p$, its goal is to compute $D_2 = \hat{e}(g,g)^v$. $\mathcal{B}$ simulates a challenger, who is allowed to access the signing oracle $\mathcal{SO}$ that can output the signatures on individual reports, and interacts with the adversary $\mathcal{A}$ as follows.

- ▷ In setup, $\mathcal{B}$ randomly chooses $r_i \in \mathbb{Z}_p$ to set the public key h_i to D^{r_i} and the re-sign key RK_i to $D_1^{r_i}$, for $1 \leq i \leq N$. Then, $\mathcal{B}$ randomly picks $\gamma \in \mathbb{Z}_p$ to set the parameter g_0 to g^γ. Finally, $\mathcal{B}$ sends $(\{h_i, RK_i\}_{1 \leq i \leq N}, g, g_0)$ to $\mathcal{A}$.
- ▷ $\mathcal{A}$ queries the signatures on individual reports under any public key in $\{h_i\}$ for $1 \leq i \leq N$. $\mathcal{B}$ issues a signature query to $\mathcal{SO}$ and receives σ_i, then, returns σ_i to $\mathcal{A}$.
- ▷ Eventually, $\mathcal{A}$ produces an aggregated signature $\hat{\sigma}$ on the compressed reports $(SM_i, \mathcal{U}, \hat{t}, \hat{a}, \hat{m})$ for $1 \leq i \leq N$ in a time period $t \in Q$. Suppose that $\hat{\sigma}$ is a valid signature on $\hat{m}$; otherwise, $\mathcal{B}$ reports failure and aborts. Thus, $\hat{\sigma}$ satisfies the verification equation, i.e.,

$$\hat{\sigma} = \hat{e}(\prod_{t \in Q} \prod_{i=1}^{N} H(SM_i || \mathcal{U} || t) g^{\hat{a}} g_0^{\hat{m}}, g^v).$$

  Assume the expected signature, which would be obtained from the honest smart meters, be σ on the report $(SM_i, \mathcal{U}, t, a, m)$ for $1 \leq i \leq N$ in a time period $t \in Q$. σ also satisfies the verification equation, i.e.,

$$\sigma = \hat{e}(\prod_{t \in Q} \prod_{i=1}^{N} H(SM_i || \mathcal{U} || t) g^{a} g_0^{m}, g^v).$$

  If $\hat{a} = a$ and $\hat{m} = m$, then $\hat{\sigma} = \sigma$. Define $\Delta a = \hat{a} - a$ and $\Delta m = \hat{m} - m$, then, either Δa or Δm is nonzero.

  - If $\hat{\sigma} \neq \sigma$, we divide the verification equation for $\hat{\sigma}$ by the equation for σ and obtain

$$\hat{\sigma}/\sigma = \hat{e}(\prod_{t \in Q} \prod_{i=1}^{N} g^{\Delta a} g_0^{\Delta m}, g^v).$$

Since $g_0 = g^\gamma$, we have

$$\hat{\sigma}/\sigma = \hat{e}(g^{\sum_{t\in Q}\sum_{i=1}^{N}\Delta a+\gamma\Delta m}, g^v).$$

Rearranging the equation yields

$$D_2 = \hat{e}(g,g)^v = (\tfrac{\sigma}{\hat{\sigma}})^{\sum_{t\in Q}\sum_{i=1}^{N}\Delta a+\gamma\Delta m},$$

which is the solution to the CONF problem.

– Otherwise, we get $\hat{e}(g^{\sum_{t\in Q}\sum_{i=1}^{N}\Delta a+\gamma\Delta m}, g^v) = 1$ and

$$D_2 = \hat{e}(g,g)^v = 1^{\sum_{t\in Q}\sum_{i=1}^{N}\Delta a+\gamma\Delta m}.$$

So we can solve the CONF problem.

Therefore, if the CONF problem cannot be solved within a non-negligible probability in probabilistic polynomial time, any adversary cannot corrupt the aggregated reports. ■

In summary, P^{2}SM can achieve the authentication, confidentiality and integrity of the individual consumption reports and aggregated reports, as well as the electricity bills. Thus, the misbehaving collector cannot corrupt both consumption reports and electricity bills without being detected.

5.5 Performance Evaluation

We evaluate the performance of P^{2}SM in terms of computational, communication and storage burden, and discuss the implementation on the current advanced metering infrastructure.

5.5.1 Computational Cost

To evaluate the computational cost of P^{2}SM, we count the number of time-consuming operations on elliptic curve groups, including scalar multiplication (SM), point addition (PA), hash to point (HP), bilinear pairing (BP) and multiplication in $\mathbb{G}_T$ (MU_T).

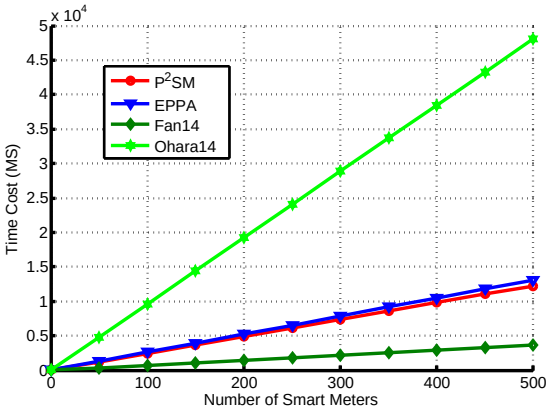
Table 5.1: Comparison of Time Costs

Unit: ms, $N = 100$

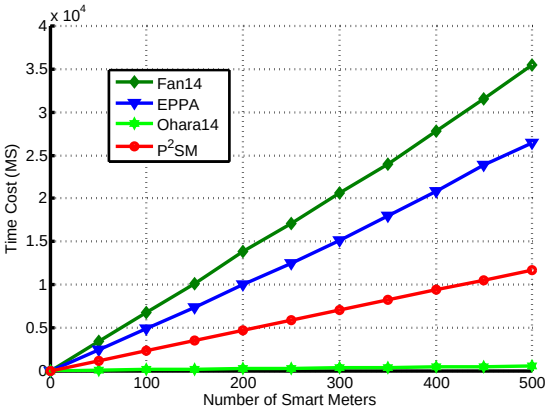
Phase	Report Generation	Report Aggregation	Report Reading
	Smart Meter	Collector	Operation Center
P^{2}SM	24.7	2355.4	328.1
EPPA [142]	26.2	4906.3	97.7
Fan14 [154]	7.3	6737.6	130.9
Ohara14 [155]	96.2	25.4	174.3

Phase	Dynamic Billing		
	Collector	Operation Center	Customer
P^{2}SM	2254.7	3564.1	54.9
EPPA [142]	Null	Null	Null
Fan14 [154]	Null	Null	Null
Ohara14 [155]	Null	26.5	183.4

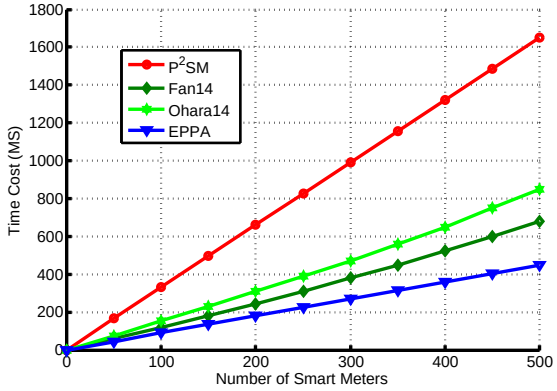
When a customer C_i registers on OC, it is required to perform $3SM{+}HP{+}2PA$ operations to generate $(\mathcal{H}_i, z_{i1}, z_{i2})$, and OC runs $2SM{+}PA$ operations to recover k_i and computes RK_i. In each reporting slot t, SM_i computes $(a'_{it}, c_{it}, t_{it}, \sigma_{it})$ to generate P_{it}, which needs SM_i to execute $4SM{+}HP{+}2PA{+}2MU_T$ operations. Here $\hat{e}(g_0, g)$ and $\hat{e}(g, g)$ can be pre-computed in System Initialization phase to reduce the computational overhead of SM_i. To aggregate the individual consumption reports, the collector performs $(2N|Q| - 2)PA + N|Q|BP$ operations to generate P, where $|Q|$ denotes the number of individual reports of SM_i received in a forwarding period Q. Finally, OC executes $(2N|Q| + 1)PA + (N|Q| + 4)SM + N|Q|HP + MU_T + BP$ and discrete logarithm operations to obtain the sum of power consumption in $\mathbb{RA}$, if all the reports are valid. Otherwise, OC needs to execute $N|Q|(2SM + PA)$ operations to find the invalid authentication messages if the equation (4) does not hold; or $N|Q|(2PA + 2SM + HP + 2BP)$ operations to identify the invalid signatures if the equation (5) does not hold. In Dynamic Billing phase, for a customer C_i, the collector needs to aggregate the power consumption by performing $72|Q_j|SM + (24|Q_j| - 3)PA$ operations, where $|Q_j|$ denotes the number of reporting slots in each forwarding slot. Then, OC runs $NSM + (2N - 1)MU_T$ operations to verify the correctness of electricity bills and executes $SM + BP$ operations to generate l_i or $\mathcal{U}$ helps OC to compute BP operation. Finally, C_i performs MU_T and discrete logarithm operations to recover the bill d_i, and checks the correctness of d_i by running $(24|j|{+}2)SM{+}2PA{+}24|j|HP{+}2BP$ operations.



(a) Computational Cost of Smart Meters

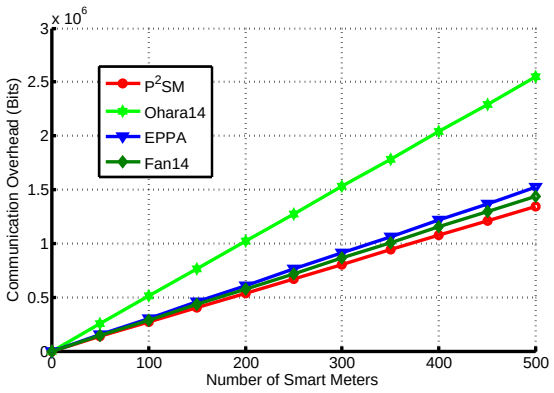


(b) Computational Cost of the Collector

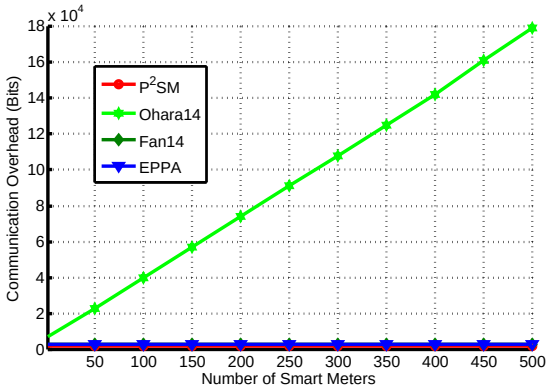


(c) Computational Cost of the Operation Center

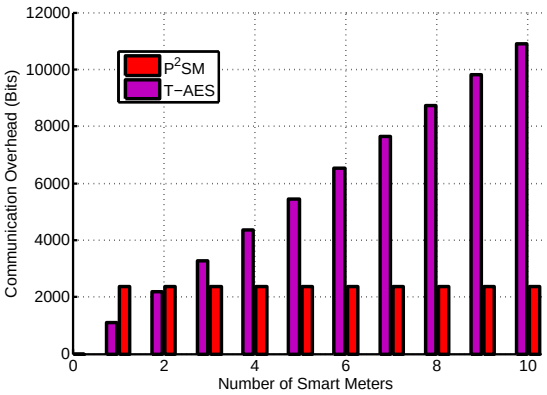
Figure 5.3: Comparison on Computational Overhead



(a) Overhead between SM and Collector



(b) Overhead between Collector and OC



(c) Comparison of P^{2}SM and TLS protocol

Figure 5.4: Comparison on Communication Overhead

We conduct an experience on a notebook with Intel Core i5-4200U CPU @ 2.29GHz and 4.00GB memory. We use the MIRACL library to implement number-theoretic based methods of cryptography. The Weil pairing is utilized to realize the bilinear pairing and the elliptic curve is chosen with a base field size of 512 bits. The parameter p is 160 bits. We compare the P^{2}SM with three schemes, EPPA [142] (based on Paillier encryption [156]), Fan14 [154] (based on BGN encryption [149]) and Ohara14 [155] (based on Lifted ElGamal encryption [152]). While P^{2}SM is constructed from proxy re-encryption [22]. To keep the consistency, we utilize the same settings of these schemes in the experience. The number of customers in $\mathbb{RA}$ is 100, the number of reporting slots in a period Q is 1 and the number of reporting period per day φ is 24. The execution time of each entity in report generation, report aggregation, report reading and dynamic billing phases of four schemes are shown in Table I. Fig. 5.3 shows the comparison results of four schemes about the computational cost for each entity. Although Fan14 [154] is the most efficient scheme in report generation, since it utilizes the BGN encryption to encrypt the meter measurements and no commitment is generated, it is the most inefficient one in report aggregation. EPPA [142] is the fastest scheme in report reading, as no discrete logarithm computation is needed and less bilinear pairings are computed compared with P^{2}SM, but it is time-consuming in report aggregation. Moreover, Fan14 [154] and EPPA [142] do not achieve dynamic billing for customers. Ohara14 [155] has a good performance on the computational overhead, but it is vulnerable to the pollution attacks from misbehaving collectors, as well as Fan14 [154] and EPPA [142]. P^{2}SM is not the most efficient one in four schemes, even the least efficient one in report reading, because it utilizes the pairing-based cryptosystem to offer a higher security guarantee compared with Fan14 [154], EPPA [142] and Ohara14 [155]. Moreover, the bottleneck of the smart metering on computational capability is the smart meters, while our scheme is very efficient on report generation for smart meters. P^{2}SM is still efficient since the time cost on report reading is only 328ms, while the operation center is always powerful on computation. The collector's computational overhead can be reduced by decreasing the number of smart meters in its coverage area and building the hierarchical network structure to improve the efficiency of meter measurement collection.

5.5.2 Communication and Storage Overhead

The communications of P^{2}SM composes of SM-to-Collector communication and Collector-to-OC communication. In the SM-to-Collector communication, SM_i is required to send 2688-bit P_{it} at a time slot t to the collector, where $\mathcal{U}$, SM_i and t are assumed to be 160 bits, respectively. In the Collector-to-OC communication, all P_{it} are aggregated to be P, which is only 2388 bits, if $\mathcal{C}$ and Q are 160 bits, respectively. Therefore, the communication

overhead is significantly reduced through data aggregation. In addition, the collector aggregates the individual reports to generate a bill for each customer every day. The bill is only 2880 bits. Fig. 5.4 shows the comparison results of four schemes about the communication overhead of SM-to-Collector communication (Fig. 5.4(a)) and Collector-to-OC communication (Fig. 5.4(b)). Since the ciphertext of proxy re-encryption [22] is shorter than those of Paillier encryption [156], BGN encryption [149] and Lifted ElGamal encryption [152] (the commitment in [152] is 1024 bits), the communication burden of SM-to-Collector communication in P^{2}SM is lower than those in the other three schemes as shown in Fig. 5.4(a). After the individual consumption reports are aggregated, the communication overhead becomes constant, which is still lower than those in EPPA [142], Fan14 [154] and Ohara14 [155] (the overhead of Collector-to-OC communication is linear to the number of smart meters in the home area network). Fig. 5.4(c) shows the comparison on communication overhead of P^{2}SM and TLS protocol (T-AES) [141], in which AES-256 is used to encrypt meter readings and BLS signature [127] is used to guarantee authentication and data integrity. If $N > 2$ in $\mathbb{RA}$, P^{2}SM is more efficient than TLS protocol on Collector-to-OC communication.

In addition, the collector needs sufficient storage space to transiently maintain the individual reports in $\mathbb{RA}$. If $N = 1000$ in $\mathbb{RA}$ and $\rho = 15$, these individual reports would possess 30.8MB storage space every day. Therefore, each collector only needs to deploy 61.6MB memory to support power consumption collection and dynamic billing for customers.

5.6 Summary

In this chapter, we have introduced a new security model to formally define the misbehavior of hacked collectors and have proposed a privacy-preserving smart metering scheme to achieve end-to-end security and high communication efficiency in smart grid. P^{2}SM not only allows collectors to aggregate authentication messages, meter readings and signatures to reduce communication overhead and preserve the privacy of customers, but also prevents a misbehaving collector from corrupting power consumption reports. Further, the collector is able to generate verifiable daily electricity bills from individual consumption reports based on dynamic prices for customers. We have proved the security of P^{2}SM and evaluated its performance through the comparison with the existing schemes. P^{2}SM is a secure and efficient communication protocol that can replace the TLS protocol to achieve secure smart metering in smart grid.

Chapter 6

Dual-anonymous Reward Distribution

6.1 Introduction

The proliferation of increasingly capable mobile devices (e.g., smartphones, smartwatches, smartglasses) with a plethora of on-board sensors (e.g., accelerometer, gyroscope, camera, GPS) has given rise to MCS [1], a compelling paradigm that enables mobile users to collect, process and share sensing data from social events and phenomena. Due to the rich resources and sensing modalities of mobile devices and the mobility and intelligence of mobile users, MCS can support large-scale sensing applications for collecting higher-quality and semantically complex data. Currently, a large variety of MCS systems (e.g., GreenGPS [157], SmartRoad [158] and Jigsaw [159]) have been emerged, which serve an ever-increasing number of sensing applications, including vehicular navigation, indoor floor reconstruction, environment monitoring, urban sensing, and many others.

Participating in such MCS applications is usually a costly activity for mobile users, since it may consume not only their time, but also system resources of mobile devices, e.g., computing power, communication bandwidth and battery [101]. Moreover, some spatial-temporal tasks need mobile users to physically go to specific locations at certain time points for data collection, such that they have to spend travel cost to reach the locations in order to perform the tasks. Therefore, without satisfactory rewards that can compensate the cost of mobile users, they will be reluctant to participate in MCS tasks. The incentivizing mobile users participation in MCS systems is paramountly important and promising to focus on. Many incentive mechanisms [54, 160, 161, 162, 163, 164] have been proposed to

encourage mobile users for participation, which can be divided into two categories, reward-sharing mechanisms [54, 161] and auction-based mechanisms [162, 163, 164]. Typically, a reward-sharing mechanism claims the reward that a customer can provide to attract mobile users for participation and distributes the reward based on the claimed rules after the task fulfillment, and an auction-based incentive mechanism chooses the mobile users who are interested in performing tasks based on their bids and determines the amount of payments to mobile users for compensating their costs. Nevertheless, these mechanisms only achieve a single design goal of incentive but ignore other critical issues.

The practical incentive mechanism also needs to preserve the privacy for both customers and mobile users, because they are unwilling to participate in crowdsensing activities if their privacy is exposed [31]. Firstly, the releasing tasks may contain some sensitive information, from which a curious attacker can learn the intentions why customers issue these tasks. For instance, if a customer releases a task to collect noise level and traffic condition in a residential area, a house agent can learn that this customer may attempt to buy a house in that area [165]. Secondly, the sensing data are necessarily people-centric and related to some aspects of mobile users and their social setting: where mobile users are and where they are going; what habits they have and what places they frequently visit; how is their health status and which activity they prefer to do. The leakage of these information may cause plenty of troubles to mobile users in life and even threats to their lives and property. For example, when a mobile user delivers the comments on a medical experience of visiting a psychologist, the service provider is aware of that this mobile user may have some mental diseases. As a result, he may suffer discrimination in life. Therefore, if there is no sufficient privacy-preserving mechanism to preserve the privacy of mobile users, they may be reluctant to participate in crowdsensing tasks, although the reward obtained from the customers can compensate their costs on data collection.

To overcome these obstacles, some privacy-preserving incentive mechanisms [91, 94, 88, 95] have been proposed to encourage mobile users to participate in crowdsensing tasks and protect the privacy of mobile users, simultaneously. Most of these [91, 94, 88] studied the auction-based incentive methods, which allow a crowdsensing server provider to select a set of anonymous mobile users to perform tasks according to their bids. However, privacy-preserving reward-sharing mechanisms have not received enough attention lately. Niu et al. [95] designed a new E-cent protocol and proposed an E-cent-based privacy-preserving incentive mechanism to achieve reward distribution. Nevertheless, this protocol cannot trace the customer who double-uses the E-cent, since all the identities of customers and mobile users are invisible. Subsequently, Zhang et al. [41] utilized dividable electrical cash to design a privacy-preserving market scheme for privacy-preserving reward sharing. However, this scheme is inefficient and the identities of mobile users would be exposed to

the curious service provider.

In this chapter, we propose a Dual-Anonymous Reward Distribution (DARD) scheme from randomizable techniques to achieve efficient and privacy-enhanced incentive in MCS systems [166]. Specifically, the main contributions of this chapter are as follows:

- ▷ We propose DARD, a new reward distribution scheme based on the PS signature [17]. In DARD, a customer creates a task and claims a fixed amount of coins withdrew from the bank as a reward for attracting mobile users to perform the task, and the service provider divides the reward into several individual rewards and distributes them to mobile users according to their contributions on the task. The mobile users deposit their obtained rewards to the bank. We design an approach of coins circulation in reward-sharing incentive mechanism and achieve the payment balance of the bank in DARD.
- ▷ We preserve the privacy of both customers and mobile users in DARD. By adopting randomizable techniques, the customer can claim the reward anonymously and every mobile user deposits the individual reward distributed by the service provider to the bank without disclosing the identity. Even the customer, colluding with the bank and the service provider, cannot learn identity information about the participating mobile users. In addition, the identity of a cheater, who double-uses the coins or rewards, can be recovered by a trusted authority.
- ▷ DARD allows the service provider to divide the claimed reward into several individual rewards with various amount of coins and distribute them to the corresponding mobile users, such that it is unnecessary for the service provider to deal with each coin separately. Therefore, the computational and communication overhead is quite low.

6.2 Problem Statement

We state the problem by formalizing system model and security model, and identifying security goals.

6.2.1 System Model

System model consists of five kinds of entities: a trusted authority (TA), a bank, a crowd-sensing service provider, customers and mobile users. The TA bootstraps the mobile

crowdsensing system and generates public key certificates for all the entities in the system. It is also responsible to detect the cheater who double-claims coins or double-distributes rewards to mobile users. The bank is a financial institution, which manages the circulation of electrical coins. It issues electrical coins to customers and receives the coins from mobile users for deposit. The crowdsensing service provider (CSP) provides mobile crowdsensing services to customers. Specifically, it assigns mobile crowdsensing tasks to mobile users, collects and aggregates crowdsensing reports, distributes rewards to mobile users according to their distinct contributions on tasks. The customers can be individuals or organizations. They have mobile crowdsensing tasks to perform but do not have sufficient resources to accomplish individually. Hence, they release these tasks on the CSP. They also withdraw electrical coins from the bank and claim the coins as a reward to attract mobile users for performing their tasks. The mobile users perform the tasks to earn coins using their own mobile devices with rich resources of data sensing, processing and communications.

A customer firstly creates a mobile crowdsensing task and withdraws electrical coins from the bank. When owning the coins, the customer determines the amount of coins used to reward participating mobile users and sends the task to the CSP, along with a claim on the reward. The CSP releases the received task and the claimed reward to attract mobile users for participation. The mobile users, who are interested in the releasing task, collect data according to the requirements of the task and report their sensing data to the CSP. After gathering enough crowdsensing reports from mobile users, the CSP generates and returns a crowdsensing result to the customer. Then, the CSP determines the amount of coins as an individual reward that a specific mobile user could obtain based on his/her contribution on the task, and distributes it to the mobile user. After receiving the individual reward from the CSP, the mobile user deposits it to the bank. As illustrated in Fig. 6.1, in the system model of DARD, we describe the processes of reward claim and distribution, coins withdraw and deposit. Task releasing and allocation, data collection and reporting are the same as those in [165].

6.2.2 Threat Model

In threat model, all the entities except TA are semi-honest, since everyone may be greedy for wealth. Specifically, a customer may double-claim the electrical coins, indicating that the customer may claim a same coin in two or more mobile crowdsensing tasks, and the CSP might distribute a coin to two or more mobile users. The mobile user may also deposit a coin to the bank more than once. These misbehavior would break the payment balance of the bank. Furthermore, although the CSP and the bank have their responsibility for hosting their business and service honestly, they are curious about customers and

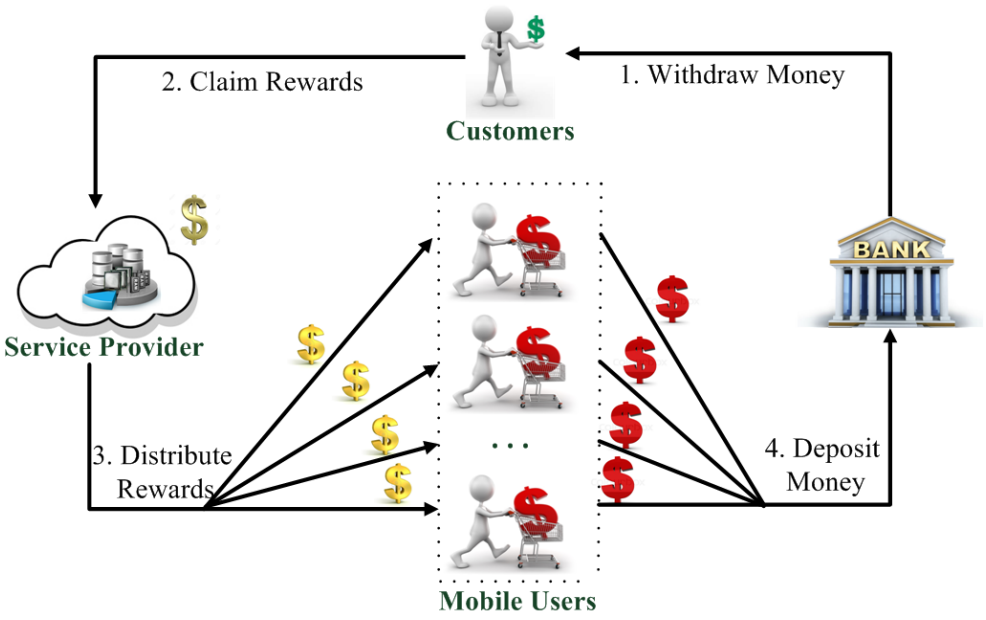


Figure 6.1: System Model of DARD

mobile users, including identity, location, habit, health status, daily route and political affiliation. The customers and mobile users are not fully trusted. They are also interested in other customers who release similar tasks or mobile users who perform the same tasks. In addition, to have a powerful attack capability, the CSP, the bank, some customers and mobile users may collude to capture the privacy of a specific customer or mobile user. Nevertheless, the majority of mobile users honestly perform the tasks based on their requirements for earning rewards.

6.2.3 Security Goals

To design a secure reward distribution scheme under the aforementioned system model and security model, DARD should achieve the following security goals.

- ▷ *Dual Anonymity*: To protect the privacy of both customers and mobile users, their identities should be preserved against curious attackers. Although the contents of a crowdsensing task or a crowdsensing report may be disclosed, it is impossible for attackers to link these contents to a specific customer or mobile user.

- ▷ *Reward Balance*: To ensure the payment balance of the bank, a customer is unable to claim the coins more than he withdrew from the bank without being detected, even he colludes with the CSP; the CSP cannot distribute the coins more than the total coins claimed by the customer without being detected. Moreover, the mobile users are not able to deposit more than the customer withdrew from the bank, even they can collude with other entities.
- ▷ *Cheater Tracing*: If a customer double-claim the withdrew coins, the TA can recover the identity of the customer. Besides, the TA can catch the misbehavior of double-distribution of the CSP, when the mobile users deposit the individual rewards to the bank.

6.3 DARD

We propose our DARD scheme, which consists of six phases: Setup, Withdraw, Claim, Distribution, Deposit and Cheater Tracing.

6.3.1 Setup

This phase is run by the TA to bootstrap the mobile crowdsening system. Given a security parameter λ, TA defines three cyclic groups $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ of a same prime order p. Let $\hat{e} : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ be a bilinear pairing of type 3, where $\mathbb{G}_1 \neq \mathbb{G}_2$ and no efficiently computable homomorphism exists between $\mathbb{G}_1$ and $\mathbb{G}_2$ in either direction, g be a generator of $\mathbb{G}_1$ and $\tilde{g}$ be a generator of $\mathbb{G}_2$. $\mathcal{H} : \{0,1\}^* \to \mathbb{Z}_p^*$ is a collision-resistant hash function. Assume that the number of coins a customer can withdraw from the bank once be $V = 2^n$, which is denoted as a wallet. Let $\mathcal{S}_n$ be the set of bitstrings of size smaller than or equal to n and $\mathcal{F}_n$ be the set of bitstrings of size exactly n. TA generates a binary tree of depth n as illustrated on Fig. 6.2. The root of the tree is an empty string ε, each node of the tree refers to an element $s \in \mathcal{S}_n$ and each leaf to an element $f \in \mathcal{F}_n$. For any node $x \in \mathcal{S}_n$, $\mathcal{F}_n(x)$ contains all the leaves in the subtree below x.

- ▷ For each leaf $f \in \mathcal{F}_n$, TA randomly picks $l_f \leftarrow \mathbb{Z}_p^*$.
- ▷ For each node $s \in \mathcal{S}_n$, TA randomly picks $r_s \leftarrow \mathbb{Z}_p^*$ to compute $g_s = g^{r_s}$.
- ▷ For each $s \in \mathcal{S}_n$ and for each $f \in \mathcal{F}_n(s)$, TA computes $\tilde{g}_{s \to f} = \tilde{g}^{l_f / r_s}$.

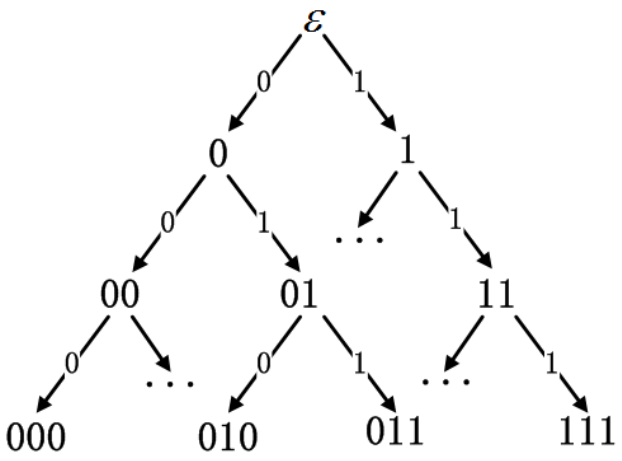


Figure 6.2: Binary Tree

The public parameters are $\{p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, \hat{e}, g, \tilde{g}, \mathcal{H}, n, V, \mathcal{F}_n, \mathcal{S}_n, \{g_s\}_{s\in\mathcal{S}_n}, \{\tilde{g}_{s\to f}\}_{s\in\mathcal{S}_n, f\in\mathcal{F}_n(s)}\}$. TA keeps $\{r_s\}_{s\in\mathcal{S}_n}$ and $\{l_f\}_{f\in\mathcal{F}_n}$ in private.

The bank randomly chooses $\alpha, \beta \leftarrow \mathbb{Z}_p^*$ to compute $\tilde{v}_1 = \tilde{g}^\alpha$ and $\tilde{v}_2 = \tilde{g}^\beta$. The secret key of the bank is (α, β) and its public key is $(\tilde{v}_1, \tilde{v}_2)$. Besides, the bank maintains two lists (L_1, L_2). L_1 keeps y_i identifying the deposited claims and L_2 stores π_i identifying the deposited individual rewards.

The CSP chooses a random $\tau \leftarrow \mathbb{Z}_p^*$ as its secret key and calculates its corresponding public key as $\hat{v} = g^\tau$.

Each customer or mobile user also selects a random $\gamma \leftarrow \mathbb{Z}_p^*$ as the secret key and computes the public key as $v = g^\gamma$.

6.3.2 Withdraw

To withdraw V from the bank, a customer firstly picks random $x, k_1, k_2 \in \mathbb{Z}_p^*$ to compute

$$\begin{array}{lll} C = g^x, & T_1 = g^{k_1}, & T_2 = g^{k_2}, \\ c = \mathcal{H}(T_1||T_2||V), & z_1 = k_1 + c\gamma, & z_2 = k_2 + cx. \end{array}$$

He then sends a withdraw request (C, c, z_1, z_2) to the bank. After receiving (C, c, z_1, z_2) from the customer, the bank computes $T_1' = g^{z_1}v^{-c}$, $T_2' = g^{z_2}C^{-c}$ and verifies whether $c = \mathcal{H}(T_1'||T_2'||V)$ holds or not and C is not previously used. If either does not hold,

the bank returns failure and aborts; otherwise, the bank chooses a random $u \in \mathbb{Z}_p^*$ and computes a signature on C as

$$\sigma_1 = g^u, \qquad \sigma_2 = (g^\alpha C^\beta)^u, \qquad \sigma_3 = \hat{e}(\sigma_1, \tilde{v}_2).$$

The bank returns the signature $(\sigma_1, \sigma_2, \sigma_3)$ to the customer. Finally, the customer keeps $M = (x, \sigma_1, \sigma_2, \sigma_3)$ as the withdrew coins with a wallet V.

6.3.3 Claim

A customer with a wallet V generates a mobile crowdsensing task TK to recruit mobile users for data collection. To encourage mobile users to participate in TK, the customer determines the number of coins $L = 2^l \leq 2^n$ as a reward to be distributed to mobile users who make contributions on fulfilling the task. To generate a claim on the reward L, the customer firstly selects an unused node s of level $n - l$ and computes $t_s = g_s^x$. The customer then randomly chooses $t \leftarrow \mathbb{Z}_p^*$ to randomize M as $\sigma_1' = \sigma_1^t$, $\sigma_2' = \sigma_2^t$, and proves that (σ_1', σ_2') is a valid signature on x and $t_s = g_s^x$. To do so, the customer selects a random $k \leftarrow \mathbb{Z}_p^*$ to compute

$$\begin{aligned} K &= g_s^k, & \sigma_3' &= \sigma_3^{kt}, \\ \mathcal{C} &= \mathcal{H}(\sigma_1' || \sigma_2' || t_s || K || \sigma_3' || TK), & \mathcal{Z} &= k + \mathcal{C}x. \end{aligned}$$

Finally, the customer sends the claim $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z})$ to CSP, along with the task TK.

Upon receiving $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z}, TK)$, the CSP learns the amount of the reward L from g_s and verifies the validity of this claim by computing

$$K' = g_s^{\mathcal{Z}} t_s^{-\mathcal{C}}, \qquad \sigma_3'' = \hat{e}(\sigma_1', \tilde{v}_1)^{\mathcal{C}} \hat{e}(\sigma_2', \tilde{g})^{-\mathcal{C}} \hat{e}(\sigma_1', \tilde{v}_2)^{\mathcal{Z}},$$

and checking whether $\mathcal{C} = \mathcal{H}(\sigma_1' || \sigma_2' || t_s || K' || \sigma_3'' || TK)$ or not. If not, the CSP returns failure and aborts; otherwise, it releases the crowdsensing task TK and recruits mobile users for performing TK. At the same time, the CSP publishes the claim $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z})$ to attract mobile users for participation.

6.3.4 Distribution

Some mobile users, who are interested in the crowdsensing task TK, collect data according to the requirements of TK and report the sensing data to the CSP. The CSP gathers all the crowdsensing reports and returns a result to the customer. After that, the SP determines the number of coins a specific mobile user U_i can obtain based on his contribution on TK and then distributes an individual reward to U_i. Suppose U_i can get $L_i = 2^{l_i}$ coins from this task, where $l_i \leq l$. To do so, the CSP chooses an unused node s_i of level $n - l_i$ below the node s, and computes $t_{s_i} = g_{s_i}^{\tau}$. The CSP also picks a random $w \in \mathbb{Z}_p^*$ to compute $W = g^w$, $e = \mathcal{H}(\sigma_1'||\sigma_2'||t_s||g_s||\mathcal{C}||\mathcal{Z}||t_{s_i}||g_{s_i}||W)$ and $f = w + e\tau$. Finally, the CSP sends the individual reward $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z}, t_{s_i}, g_{s_i}, e, f)$ to U_i.

When receiving the individual reward from the CSP, U_i verifies whether the obtained reward is valid or not. Specifically, U_i checks whether s_i is a node of the subtree below s, and verifies the CSP's signature by computing $W^* = g^f \hat{v}^{-e}$ and checking whether $e = \mathcal{H}(\sigma_1'||\sigma_2'||t_s||g_s||\mathcal{C}||\mathcal{Z}||t_{s_i}||g_{s_i}||W^*)$ or not. U_i also checks the validity of $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z})$. If one of three conditions does not hold, U_i returns failure and aborts; otherwise, U_i accepts the individual reward $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z}, t_{s_i}, g_{s_i}, e, f)$.

6.3.5 Deposit

U_i forwards the individual reward to the bank for deposit. Upon receiving $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z}, TK, t_{s_i}, g_{s_i}, e, f)$, the bank firstly checks that it is not previously used and verifies its validity. Since the bank can know the used node s from g_s, for each leaf $f \in \mathcal{F}_n(s)$, it computes $y_j = \hat{e}(t_s, \tilde{g}_{s \to f})$ and checks whether $y_j \in L_1$. If $\forall j$, $y_j \notin L_1$, then the bank verifies the validity of the claim and adds these elements $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z}, TK)$ to the list L_1 if the claim is valid. Else, there is an element $\bar{y} \in L_1$ such that $\bar{y} = y_i$. The bank finds the corresponding claim $(\bar{\sigma}_1', \bar{\sigma}_2', \bar{t}_s, \bar{g}_s, \overline{\mathcal{C}}, \overline{\mathcal{Z}}, \overline{TK})$ from L_1 and sends $(\bar{\sigma}_1', \bar{\sigma}_2', \bar{t}_s, \bar{g}_s, \overline{\mathcal{C}}, \overline{\mathcal{Z}}, \overline{TK})$ and $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z}, TK)$ to TA. The bank also learns the used node s_i from g_{s_i}, for each leaf $f \in \mathcal{F}_n(s_i)$, it calculates $\pi_\rho = \hat{e}(t_{s_i}, \tilde{g}_{s_i \to f})$ and checks whether $\pi_\rho \in L_2$. If $\forall \rho$, $\pi_\rho \notin L_2$, then the bank verifies the validity of the CSP's signature (e, f) and adds these elements $(t_s, g_s, t_{s_i}, g_{s_i}, e, f)$ to the list L_2 if the signature is valid. Else, there is an element $\bar{\pi} \in L_2$ such that $\bar{\pi} = \pi_\rho$. The bank finds the corresponding $(\bar{t}_s, \bar{g}_s, \bar{t}_{s_i}, \bar{g}_{s_i}, \bar{e}, \bar{f})$ from L_2, and sends $(\bar{t}_s, \bar{g}_s, \bar{t}_{s_i}, \bar{g}_{s_i}, \bar{e}, \bar{f})$ and $(t_s, g_s, t_{s_i}, g_{s_i}, e, f)$ to TA. If the individual reward is valid and unused previously, the bank returns 1 to U_i. Then, U_i randomly chooses $x_1^*, k_1^*, k_2^* \in \mathbb{Z}_p^*$ to compute

$$C^* = g^{\gamma_i} g_{s_i}^{x_1^*}, \qquad T^* = g^{k_1^*} g_{s_i}^{k_2^*},$$

$$c^* = \mathcal{H}(T^*||L_i), \quad z_1^* = k_1^* + c^*\gamma_i, \quad z_2^* = k_2^* + c^* x_1^*.$$

U_i sends $(C^*, c^*, z_1^*, z_2^*, L_i)$ to the bank. The bank verifies the validity of (C^*, c^*, z_1^*, z_2^*) by computing $T'' = g^{z_1^*} g_{s_i}^{z_2^*} (C^*)^{-c^*}$ and checking whether $c^* = \mathcal{H}(T''||L_i)$. If it does not hold, the bank returns failure and aborts; otherwise, it randomly chooses $x_2^*, h^* \in \mathbb{Z}_p$ to compute $\epsilon = (C^* g_{s_i}^{x_2^*})^{\frac{1}{\alpha + h^*}}$ and sends (x_2^*, h^*, ϵ) to U_i. After that, U_i can deposit 2^{l_i} coins to the bank by computing $x^* = x_1^* + x_2^*$ and generating a zero-knowledge proof as

$$\mathcal{SPK}\left\{(\epsilon, x^*, h^*, \gamma_i) : \begin{array}{c} v = g^{\gamma_i} \wedge \\ \hat{e}(\epsilon, \tilde{v}_1 \tilde{g}^{f^*}) = \hat{e}(g^{\gamma_i} g_{s_i}^{x^*}, \tilde{g}) \end{array}\right\}(L_i).$$

U_i sends $\mathcal{SPK}$ to the bank, along with L_i. Finally, the bank verifies the validity of $\mathcal{SPK}$ and deposits the reward for U_i if $\mathcal{SPK}$ is valid; otherwise, it aborts and returns failure.

6.3.6 Cheater Tracing

When receiving $(\bar{\sigma}_1', \bar{\sigma}_2', \bar{t}_s, \bar{g}_s, \overline{\mathcal{C}}, \overline{\mathcal{Z}}, \overline{TK})$ and $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z}, TK)$ from the bank, the TA computes $C = t_s^{t_s^{-1}}$, from which the TA can learn the identity of the customer. When receiving $(\bar{t}_s, \bar{g}_s, \bar{t}_{s_i}, \bar{g}_{s_i}, \bar{e}, \bar{f})$ and $(t_s, g_s, t_{s_i}, g_{s_i}, e, f)$ from the bank, the TA computes $\tau = t_{s_i}^{t_s^{-1}}$, which is the public key of the CSP.

6.4 Security Analysis

We explain the achieved security goals described in 6.2.3, including dual anonymity, reward balance and cheater tracing.

Dual Anonymity: To hide the identity of the customer, the bank uses the PS signature [17] to generate electronic coins. It signs the customer's commitment C to obtain $(\sigma_1, \sigma_2, \sigma_3)$, which can be randomized in the Claim phase to generate a claim $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z})$ on the reward L. Thus, the CSP and mobile users cannot learn the identity of the customer except the validity of the claim. Every mobile user receives the individual reward from the CSP and deposits it to the bank without disclosing the identity information. Specifically, U_i sends the crowdsensing report to the CSP and the CSP returns the individual reward $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z}, TK, t_{s_i}, g_{s_i}, e, f)$. It is unnecessary for U_i to expose the identity to the CSP, since the CSP distributes the individual reward based on the worth of the report.

U_i then sends $(\sigma'_1, \sigma'_2, t_s, g_s, \mathcal{C}, \mathcal{Z}, TK, t_{s_i}, g_{s_i}, e, f)$ to the bank for deposit. The bank cannot infer any personal information of U_i from $(\sigma'_1, \sigma'_2, t_s, g_s, \mathcal{C}, \mathcal{Z}, TK, t_{s_i}, g_{s_i}, e, f)$ as it is randomized from the claim. Finally, U_i proves the validity of the signature (ϵ, x^*, h^*) employing the zero-knowledge proof and stores L_i coins to the bank. Since the transcript of the zero-knowledge proof $\mathcal{SPK}$ is unlinkable to (ϵ, x^*, h^*), the bank is not able to link U_i's identity to the deposited coins, even it colludes with the CSP and the customer. Therefore, dual anonymity is achieved in DARD.

Reward Balance: To keep the payment balance of the bank, the digital signatures are adopted to guarantee that no attacker can forge or double-use the electrical coins or rewards, even the CSP, customers and mobile users are allowed to collude. The withdrew wallet is a PS signature [17], which is proved to be unforgeable based on Modified LRSW assumption. The customer can claim on $L \leq V$ to reward the participating mobile users by randomizing $(\sigma_1, \sigma_2, \sigma_3)$. The bank checks whether the withdrew coins are double-used by computing y_i and checking whether $y_j \in L_1$ for each leaf in binary tree. The CSP uses a Schnorr signature on the node s_i to generate a signature (t_{s_i}, g_{s_i}, e, f). The banks verifies whether the CSP double-uses the claimed coins to distribute individual rewards or not. Therefore, the customer and the CSP cannot double-use the withdrew coins or claimed coins without being detected by the bank, respectively. U_i also cannot double-deposit the same individual reward to the bank, since the bank can be easy to find that the individual reward received from a mobile user is the same as that stored in its database. Therefore, as the PS signature and the Schnorr signature cannot be forged and the customer, the CSP or the mobile users are unable to double-use their received coins or rewards without being detected, the payment balance of the bank can be achieved.

Cheater Tracing: The bank checks whether the claimed reward and the individual rewards are double-used by computing y_j and π_ρ, and checking whether $y_j \in L_1$ for $\forall j$, $\pi_\rho \in L_2$ for $\forall \rho$, respectively. The TA can recover the commitment of a customer $C = t_s^{t_s^{-1}}$ and the public key of the CSP $\tau = t_{s_i}^{t_s^{-1}}$, if they double-use the coins in the rewards.

6.5 Performance Evaluation

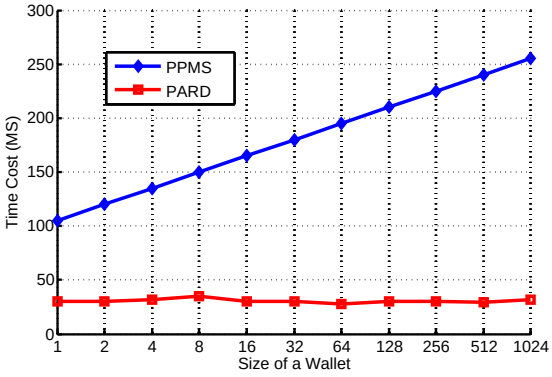
We evaluate the computational overhead of DARD by counting the number of the exponentiation in $\mathbb{G}_1$, the exponentiation in $\mathbb{G}_2$, and the bilinear pairing operations. The other operations, such as the multiplication in $\mathbb{G}_1$ and the hash operation, are not time-consuming comparing with the three kinds of operations above. We use $E_{\mathbb{G}_1}$, $E_{\mathbb{G}_2}$ and BP to denote the exponentiation in $\mathbb{G}_1$, the exponentiation in $\mathbb{G}_2$, and the bilinear pairing, respectively. In Setup phase, the TA should perform $(2^{n+1} - 1)E_{\mathbb{G}_1}$ to bootstrap the system.

The bank, the CSP and a customer/mobile user have to run $2E_{\mathbb{G}_2}$, $E_{\mathbb{G}_1}$ and $E_{\mathbb{G}_1}$ to generate their secret-public key pairs, respectively. To withdraw a wallet with V coins, a customer performs $3E_{\mathbb{G}_1}$ to generate (C, c, z_1, z_2) and the bank runs $7E_{\mathbb{G}_1} + BP$ to verify the validity of (C, c, z_1, z_2) and generates $(\sigma_1, \sigma_2, \sigma_3)$. The customer randomizes $(\sigma_1, \sigma_2, \sigma_3)$ to generate the claim by executing $5E_{\mathbb{G}_1}$ and the CSP runs $5E_{\mathbb{G}_1} + 3BP$ to verify the validity of the claim. After that, the CSP performs $2E_{\mathbb{G}_1}$ to generate an individual reward for U_i and U_i has to run $7E_{\mathbb{G}_1} + 3BP$ to check the validity of the individual reward. In the Deposit phase, the bank firstly verifies the validity of the individual reward and checks whether the claimed coins are double-used by performing $7E_{\mathbb{G}_1} + (3 + 2^l + 2^{l_i})BP$. Then, U_i runs $4E_{\mathbb{G}_1}$ to generate a new commitment (C^*, c^*, z_1^*, z_2^*) and the bank executes $5E_{\mathbb{G}_1}$ to obtain (x_2^*, h^*, ϵ). Finally, U_i performs $6E_{\mathbb{G}_1} + 4BP$ to generate $\mathcal{SPK}$ and the bank verifies it by performing $6E_{\mathbb{G}_1} + 4BP$.

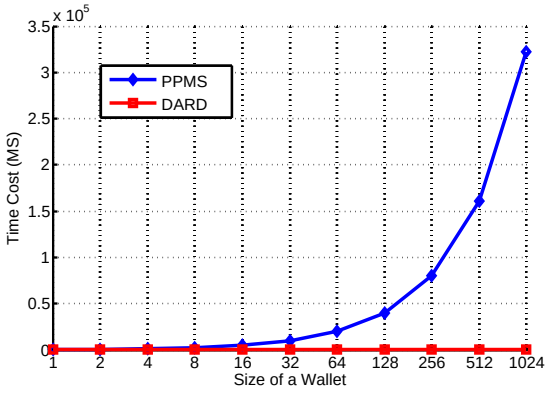
We conducted an implementation on a notebook with Intel Core i5-4200U CPU @2.29GHz and 4.00 GB memory. We use the MIRACL library to implement number-theoretic based methods of cryptography. The parameter p is 160 bits. We compare the simulation results with PPMS [167], which is a reward distribution scheme derived from dividable electrical cash. The wallet size is $V = 2^n$, where $n \in \{1, 2, \cdots, 10\}$. The computational overhead of a customer to withdraw a wallet is linear with n in PPMS, while the time cost is constant in DARD in Fig. 6.3(a). As shown in Fig. 6.3(b) and Fig. 6.3(c), the computational burden on U_i to verify the validity of the individual reward and the time cost of a customer to generate a claim are light and constant in DARD, but U_i and the customer in PPMS have to perform 2^n times to verify the individual reward and generate a claim, respectively. The computational overhead on a bank in Deposit phase of PPMS is much heavier than that in DARD in Fig 6.3(d). The reason DARD is more computation-efficient than PPMS is that 2^n coins have to be processed one by one in PPMS, but in DARD, they can be processed together.

We also discuss the communication overhead among the bank, the CSP, the customer and the mobile users. To withdraw a wallet, the customer sends (C, c, z_1, z_2) to the bank, which is 992 bits, and the bank returns 2048-bit $(\sigma_1, \sigma_2, \sigma_3)$ to the customer. In Claim phase, the customer sends a claim on 2^l coins, $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z}, TK)$, to the CSP, which is 2368+$|TK|$ bits, where $|TK|$ denotes the binary length of the task TK. Then, the CSP distributes the individual reward on 2^{l_i} coins, $(\sigma_1', \sigma_2', t_s, g_s, \mathcal{C}, \mathcal{Z}, t_{s_i}, g_{s_i}, e, f)$, to U_i, whose binary length is 3712 bits. After that, U_i forwards it to the bank for deposit. If the reward is valid, U_i sends $(C^*, c^*, z_1^*, z_2^*, L_i)$ to the bank, which is 1152 bits, and the bank responds 832-bit (x_2^*, h^*, ϵ) to U_i. Finally, U_i forwards $\mathcal{SPK}$ with the length of 3552 bits to the bank.

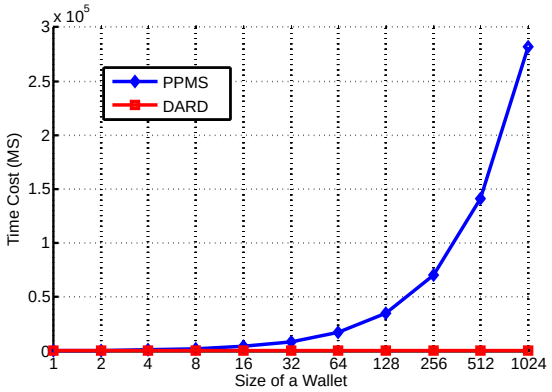
Finally, we compare the communication overhead between DARD and PPMS [41] in Fig. 6.4. In Claim and Deposit phase, the communication burden of DARD is constant,



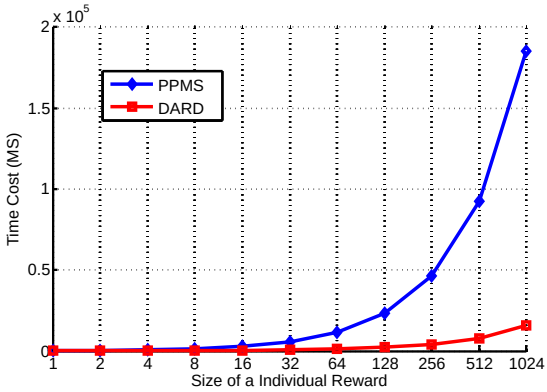
(a) Time Cost of a Customer in Withdraw



(b) Time Cost of U_i in Distribution

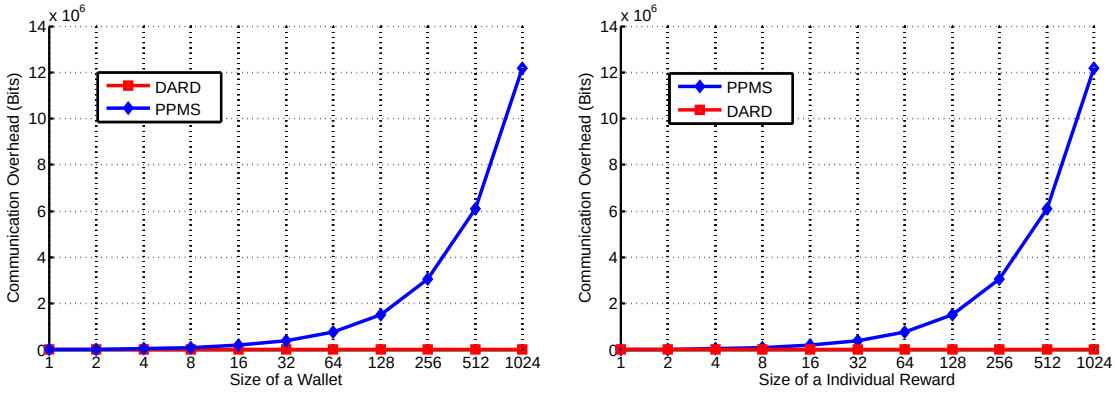


(c) Time Cost of a Customer in Claim



(d) Time Cost of the Bank in Deposit

Figure 6.3: Comparison on Computational Overhead



(a) Communication Overhead in Claim (b) Communication Overhead in Deposit

Figure 6.4: Comparison on Communication Overhead

although the size of a wallet and the number of coins in an individual reward increase exponentially. However, the communication cost of PPMS increases linearly with the size of a wallet and the number of coins in an individual reward, respectively.

6.6 Summary

We have proposed an efficient dual-anonymous reward distribution scheme to achieve reward-sharing incentive and prevent privacy leakage for customers and mobile users in MCS systems. Mainly based on the PS signature and randomizable techniques, DARD allows a customer to claim electrical coins withdrew from the bank as a reward to motivate mobile users for participating in a crowdsensing task. The mobile users can deposit the rewards awarded by the CSP to the bank without exposing their identities. Nevertheless, once the customer double-claims coins, the identity can be recovered by a trusted authority. In addition, DARD has lighter computational and communication overhead compared with the existing schemes.

Chapter 7

Conclusions and Future Work

In this thesis, we have investigated security and privacy for MCS. We highlight the main contributions of this thesis and introduce several future research directions.

7.1 Summary

In this thesis, we have developed a set of efficient, secure and privacy-preserving schemes to countermeasure and mitigate the aforementioned security and privacy threats. The requirements of security, privacy and fairness are satisfied by solving all the challenges in task allocation, data collection, data analysis and reward feedback. We summarize the following contributions of this thesis.

- To achieve privacy preservation and task allocation, we have proposed a strong privacy-preserving mobile crowdsensing scheme supporting location-based task allocation, decentralized trust management and privacy preservation for both mobile users and customers simultaneously. Based on blind signatures and randomizable matrix multiplication, the user's privacy information, including identity, location, trust level are well protected, and the customer's identity is also hidden against the service provider, mobile users and other outsiders. In doing so, the privacy of both mobile users and customers are preserved. Moreover, a privacy-preserving location matching mechanism is developed based on matrix multiplication to enable the service provider to learn whether the sensing areas of tasks and the location of mobile users are matched. To achieve credit management, we have designed a privacy-preserving credit management mechanism for mobile users, in which mobile users are

able to prove their trustworthiness without the exposure of credit points. As a result, the challenge on privacy-preserving task allocation is solidly solved that the service provider can recruit mobile users based on the requirements of crowdsensing tasks, the locations and trust levels of mobile users without invading the privacy of either mobile users or customers.

- To enable data confidentiality and data collection, we have proposed a fog-assisted mobile crowdsensing framework that allows fog nodes to perform task allocation and data deduplication, and developed a fog-assisted secure data deduplication scheme to achieve the detection of replicate data. Based on the BLS signature, the BLS-OPRF scheme enables fog nodes to detect and delete the identical sensing data in sensing reports for saving communication bandwidth, without have any knowledge about the sensing data. To record the contribution of mobile users who generate reduplicate data with others, we have leveraged key homomorphic signature to support the aggregation of crowdsensing reports, while keeping the communication efficiency in data reporting. Moreover, We have considered the privacy leakage of mobile users from duplicate reports. To balance the trade-off between data deduplication and privacy preservation against "duplicate-linking" leakage, we have further protected the identities of mobile users who submit the identical reports with others by utilizing the blind signature, and prevent the anonymous mobile users from double-reporting sensing data and double-retrieving rewards by exploiting Chameleon hash function. The security analysis and performance evaluation demonstrate that our proposed schemes are sufficiently secure against all potential attackers and efficient with respect to the communication overhead due to the achievement of data deduplication.

- To settle the problem of privacy preservation and aggregate statistics, we investigate privacy-preserving data statistics on crowdsensed meter readings from smart meters in smart grid. We define a new security model to formalize the misbehavior of collectors, in which the misbehaving collectors can launch pollution attacks to corrupt the crowdsensed consumption during transmission. The corruption of the crowdsensed usage data may result in the false estimation of smart grid state. To prevent pollution attacks, we propose a novel privacy-preserving data statistics scheme on the crowdsensed data collected by multiple smart meters. It achieves end-to-end security, data aggregation and integrity protection against the misbehaving collectors, which act as local gateways to collect and aggregate usage data and forward to operation centers. As a result, the misbehaving collectors cannot access or corrupt power usage data of customers. In addition, a privacy-preserving verifiable linear statistics mechanism is developed to realize the linear aggregation of multiple crowdsensed

data and the verification on the correctness of aggregate results. By leveraging this mechanism, verifiable dynamic billing can be supported based on individual power consumption, and the correctness of daily bills can be verified by the customers to prevent cheating of utility. The proposed scheme achieves secure smart metering and verifiable dynamic billing against misbehaving collectors with low computational and communication overhead.

- To support privacy preservation and reward feedback, we have developed a dual-anonymous reward distribution scheme from randomizable technique to achieve efficient and privacy-enhanced incentive in MCS. In this scheme, we have modeled the reward circulation for reward-sharing incentive in MCS, in which a customer claims a fixed amount of coins withdrew from the bank as a reward for attracting mobile users in task allocation, the service provider divides the coins into several individual rewards and distributes to mobile users based on their respective contributions on the task fulfillment, and finally each mobile user is able to deposit the received reward to the bank. In doing so, the balance of rewards during coin circulation is guaranteed. Furthermore, to preserve the privacy, all the processes of coin circulation will not reveal the identities of both mobile users and customers, including reward claiming, separation, distribution and deposit. Even the bank cannot identify the coins and their owners who participate in mobile crowdsensing activities. In addition, the identity of a cheater, who double-uses the coins, can be recovered by a trusted authority. Finally, we have demonstrated that the computational and communication overhead is pretty low since the service provider is unnecessary to separately handle each coin. Instead, the service provider can divide the claimed reward into several individual rewards with various amounts of coins and distribute them to the corresponding mobile users.

7.2 Future Research Directions

This thesis introduces the MCS architecture and applications, identifies security and privacy challenges in MCS, and proposes several promising solutions to achieve security and privacy threats. Although some preliminary results on security and privacy in MCS are provided, there are still several open research directions including but not limited to the followings.

7.2.1 Blockchain-based Fair Crowdsensed Data Sharing

We have proposed a preliminary work to explore the potentials of exploiting blockchain to balance the fairness among mobile users and customers. In the future work, we further leverage the appealing properties, including distribution, opening, permanency, security, timing and pseudonymity, to extend the security, privacy and fairness in MCS. Specifically, as a decentralized and distributed digital ledger, blockchain can be used to record transactions (e.g., contracts, reputation and crowdsensed data) of mobile users and customers across many nodes so that the record cannot be altered retroactively without the corruption of all subsequent blocks and the collusion of the network. It also allows the participants to verify and audit all the on-block transactions. Further, the blockchain is managed by network nodes and serves as a distributed timestamping server, such that all the transactions are recorded with permanent timestamps. The smart contract is not only an auto-executed protocol offering fairness to participants, but a computer program that digitally facilitate, verify, or enforce the negotiation or performance of a contract with intelligence. Although the blockchain is attractive in MCS, there are still fundamental problems to be solved, such as how to build a private blockchain using proof of stake for saving power consumption, while guaranteeing the distributed and decentralized system to have the desired features, even when various participants have their individual incentives; how to provide the verification, correctness and completeness of smart contracts for enforcing the desired behaviors and abandon the irrational results. To address these issues, we will explore new techniques to build incentives for decentralized applications and design new approaches for verifying the correct execution of smart contracts.

In MCS, we plan to investigate on the approach of utilizing blockchian to achieve crowdsensed data sharing among different crowdsensing tasks. The data collected for a specific task can contribute to other related tasks for different service providers, such that it is possible to reduce the repeated work for mobile users cross different tasks. The crowdsensed data feed is promising to reduce the cost on data collection and improve data utility. However, to prevent the privacy leakage of mobile users and customers, it is unrecommendable to immediately expose all the crowdsensed data to the service provider with the purpose of data sharing. From the perspective of mobile users, the incentive of data sharing is a new problem in MCS that has not mentioned. A straightforward approach is to seek a favor from the smart contract, in which the mobile user defines the condition on the data exposure and utilization in other tasks. Nevertheless, how to determine whether the crowdsensed data for a task can be cross-used for other tasks is still challenging if the data is protected against privacy leakage. To address this problem, we may need to use privacy-preserving knowledge discovery or searchable encryption for

data matching, but the efficiency and scalability are hard to be guaranteed, once the size of the crowdsensed data is large. In addition, although smart contracts can define sharing policy of the crowdsensed data, intelligent punishment mechanisms should be designed to support the fairness of multi-party data exchange with the validation and verification in MCS. Therefore, how to define the transactions and smart contracts between participants for data sharing in MCS is the problem we will focus on in the future work.

7.2.2 Local Differentially Private Truth Discovery

In MCS, the aggregation of crowdsensed data is essential for the success of MCS. To complete crowdsensing tasks, mobile users offer the collected data that may be conflict with others' or contain noisy information. These information biased with the data provided by others may mislead customers to make false decision. To reduce the impact of false collected data, voting and aggregation are widely used approaches in MCS, but these methods cannot get rid of the negative effects. Intuitively, customers should trust the mobile users who participate in data collection and believe the crowdsensing results given by the service provider. However, users' reliability degrees are various and unknown in priori. To capture the trustworthy crowdsensed data from reliable mobile users, truth discovery is of significant importance for discovering the principle or reliable information from massive collected data. Although a large number of truth discovery schemes have been on hand, they perform the discovery operation on phaintexts, which means that the service provider has all the clear information over crowdsensed data, resulting in the privacy violation of mobile users. A handful of schemes enable privacy-preserving truth discovery in MCS based on homomorphic encryption, which end up with the heavy computational overhead for each mobile user. Differential privacy, which provides means to maximize the accuracy of queries from statistical databases, while minimizing the possibility of identifying individual record, has huge potential to be exploited for protecting individual crowdsensed data in data analysis. In MCS, each mobile user is required to keep data confidentiality before uploading them to the service provider, such that we need the "local model" of differential privacy, i.e., location differential privacy, to support the local data protection and remote data statistics. Therefore, we aim to explore efficient truth discovery by leveraging local differential privacy in MCS. Considering the trade-off between accuracy of trust discovery and the level of local differential privacy, our objective is to develop a local differentially private truth discovery scheme with compact accuracy of trust discovery and high level of privacy pretection in MCS.

7.2.3 Secure Machine Learning on Crowdsensed Data

Machine learning and data analytics perform data statistics and build models from (sensitive) crowdsensed data, such as location, medical, perference and financial data. With the popularity of ubiquitous sensing and virtual assistants, crowdsensed data feed receives collected data submission at anytime and from anywhere, such that users' privacy is at ever-increasing risk. Machine learning is powerful to discover knowledge from a large volume of data, but at the same time may learn more personal information from them. Machine learning is a double-edged sword, i.e., it extends human's capability of knowledge acquisition, invade users' privacy by exposing potential sensitive information. If the privacy does not protected, mobile users may refuse to contribute their data to crowdsensed data feed. Therefore, how to enable the utility of machine learning, while preserving user's privacy, is quite necessary to the flourish of MCS. To resolve this issue, we will further explore the privacy-enhancing techniques, such as secure multi-party computation and homomorphic encryption, to enable privacy-preserving machine learning and data analytics in the real world, and aim to design and develop a general framework to enable automatic data analytics and query analysis. Our objective is to provide practical real-world solutions for privacy-preserving machine learning and data analytics and deepen the theoretical understanding for crowdsensed data.

Increasing number of data is being collected in all domains, ranging from business activities, social networks, smart homes, intelligent transportation, to smart cities, with the promise to improve decision making and enhance human's convenience. The deep learning, such as neural networks, has expressed the huge advances and power in many application areas in many data-driven tasks. Nonetheless, deep learning can be fragile and easily fooled. For instance, an attacker could insert adversarial perturbations invisible to human vision into an image to mislead neural network to misclassify the perturbed image. Recent physical-world attacks [168] on a real stop sign cause targeted misclassification in 100% of the images in lab settings, and in 84.8% of the captured video frames obtained on a moving vehicle (field test) for the target classifier, with perturbation in the form of only black and white stickers. However, secure deep learning is in its infancy and many fundamental problems have not been solved. Why are the neural networks easily fooled? How to model the physical-world attacks on adversarial perturbation insertion? How to defense these attacks to secure neural networks? Can we build provable security theory against these physical-world attacks? Security will be one of the biggest challenges in deploying artificial intelligence, directly impacting its success in applications. Traditional program verification techniques cannot effectively secure deep learning systems. The security protection on neural networks is still open problem. Therefore, we aim to take a multi-pronged approach

to explore deeper understanding of attacks, defenses, and methods for reasoning about the security of deep learning systems.

7.3 Final Remarks

In this thesis, we have presented a suite of security and privacy-preserving schemes for mobile crowdsensing, and identified three further research directions to encourage successive research efforts and complement of this thesis. To facilitate our research accomplishments and findings to benefit real applications, we will carry out experiments to further confirm our research findings.

References

[1] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges." *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[2] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," *ACM Computing Surveys*, vol. 48, no. 1, p. 7, 2015.

[3] J. Surowiecki, *The wisdom of crowds: why the many are smarter than the few and how collective wisdom shapes business, economics, society and nations*, 2004.

[4] A. Doan, R. Ramakrishnan, and A. Y. Halevy, "Crowdsourcing systems on the world-wide web," *Communications of the ACM*, vol. 54, no. 4, pp. 86–96, 2011.

[5] J. Goldman, K. Shilton, J. Burke, D. Estrin, M. Hansen, N. Ramanathan, S. Reddy, V. Samanta, M. Srivastava, and R. West, "Participatory sensing: A citizen-powered approach to illuminating the patterns that shape our world," *Foresight & Governance Project*, pp. 1–15, 2009.

[6] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-air: when urban air quality inference meets big data," in *Proc. of ACM SIGKDD*, 2013, pp. 1436–1444.

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

[8] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. Shen, "Security and privacy in s-mart city applications: Challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017.

[9] J. Ni, K. Zhang, X. Lin, Y. Yu, and X. Shen, "Cloud-based privacy-preserving parking navigation through vehicular communications," in *Proc. of SecureComm*, 2016, pp. 85–103.

[10] J. Ni, X. Lin, K. Zhang, and X. Shen, "Privacy-preserving real-time navigation system using vehicular crowdsourcing," in *Proc. of VTC-Fall*, 2016, pp. 1–5.

[11] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi *et al.*, "Vehicular communication networks in automated driving era," *arXiv preprint arXiv:1805.09583*, 2018.

[12] J. Ni, X. Lin, and X. Shen, "Privacy-preserving data forwarding in vanets: A personal-social behavior based approach," in *Proc. of Globecom*, 2017, pp. 1–6.

[13] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. of ACM SenSys*, 2008, pp. 323–336.

[14] K. Zhang, X. Liang, J. Ni, K. Yang, and X. Shen, "Exploiting social network to enhance human-to-human infection analysis without privacy leakage," *IEEE Transactions on Dependable and Secure Computing*, no. 4, pp. 607–620, 2018.

[15] V. W. Zheng, Y. Zheng, X. Xie, and Q. Yang, "Towards mobile intelligence: Learning from gps history data for collaborative recommendation," *Artificial Intelligence*, vol. 184, pp. 17–37, 2012.

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

[17] D. Pointcheval and O. Sanders, "Short randomizable signatures," in *Proc. of CT-RSA*, 2016, pp. 111–126.

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

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

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

[21] C.-H. Li and J. Pieprzyk, "Conference key agreement from secret sharing," in *Proc. of ACISP*, 1999, pp. 64–76.

[22] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[23] M. H. Au, W. Susilo, and Y. Mu, "Constant-size dynamic k-taa," in *Proc. of SCN*, 2006, pp. 111–125.

[24] M. Bellare and O. Goldreich, "On defining proofs of knowledge," in *Proc. of Crypto*, 1992, pp. 390–420.

[25] S. Goldwasser, S. Micali, and C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on computing*, vol. 18, no. 1, pp. 186–208, 1989.

[26] I. Damgård, "Efficient concurrent zero-knowledge in the auxiliary string model," in *Proc. of EUROCRYPT*, 2000, pp. 418–430.

[27] S. Underwood, "Blockchain beyond bitcoin," *Communications of the ACM*, vol. 59, no. 11, pp. 15–17, 2016.

[28] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[29] J. B. A. M. J. Clark, A. N. J. A. K. Edward, and W. Felten, "Research perspectives and challenges for bitcoin and cryptocurrencies," *url: https://eprint. iacr. org/2015/261. pdf*, 2015.

[30] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek, "Secure multiparty computations on bitcoin," in *Proc. of IEEE S&P*, 2014, pp. 443–458.

[31] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and privacy in mobile crowdsourcing networks: challenges and opportunities," *IEEE Communications Magazine*, vol. 53, no. 8, pp. 75–81, 2015.

[32] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, "Anonysense: privacy-aware people-centric sensing," in *Proc. of Mobisys*, 2008, pp. 211–224.

[33] K. L. Huang, S. S. Kanhere, and W. Hu, "A privacy-preserving reputation system for participatory sensing," in *Proc. of LCN*, 2012, pp. 10–18.

[34] P. Korshunov, S. Cai, and T. Ebrahimi, "Crowdsourcing approach for evaluation of privacy filters in video surveillance," in *Proc. of MM*, 2012, pp. 35–40.

[35] T. Dimitriou, I. Krontiris, and A. Sabouri, "Pepper: a queriers privacy enhancing protocol for participatory sensing," in *Proc. of SecureComm*, 2012, pp. 93–106.

[36] D. Christin, C. Roßkopf, M. Hollick, L. A. Martucci, and S. S. Kanhere, "Incognisense: An anonymity-preserving reputation framework for participatory sensing applications," *Pervasive and mobile Computing*, vol. 9, no. 3, pp. 353–371, 2013.

[37] L. Kazemi and C. Shahabi, "Geocrowd: enabling query answering with spatial crowdsourcing," in *Proc. of ACM SIGSPATIAL GIS*, 2012, pp. 189–198.

[38] E. De Cristofaro and C. Soriente, "Extended capabilities for a privacy-enhanced participatory sensing infrastructure (pepsi)," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 12, pp. 2021–2033, 2013.

[39] F. Günther, M. Manulis, and A. Peter, "Privacy-enhanced participatory sensing with collusion resistance and data aggregation," in *Proc. of CNS*, 2014, pp. 321–336.

[40] D. Christin, J. Guillemet, A. Reinhardt, M. Hollick, and S. S. Kanhere, "Privacy-preserving collaborative path hiding for participatory sensing applications," in *Proc. of MASS*, 2011, pp. 341–350.

[41] Q. Ma, S. Zhang, T. Zhu, K. Liu, L. Zhang, W. He, and Y. Liu, "Plp: Protecting location privacy against correlation analyze attack in crowdsensing," *IEEE Transactions on Mobile Computing*, to appear.

[42] J. Sun, R. Zhang, X. Jin, and Y. Zhang, "Securefind: Secure and privacy-preserving object finding via mobile crowdsourcing," *IEEE Transactions on Wireless Communications*, vol. 15, no. 3, pp. 1716–1728, 2016.

[43] X. O. Wang, W. Cheng, P. Mohapatra, and T. Abdelzaher, "Enabling reputation and trust in privacy-preserving mobile sensing," *IEEE Transactions on Mobile Computing*, vol. 13, no. 12, pp. 2777–2790, 2014.

[44] H. Li, K. Jia, H. Yang, D. Liu, and L. Zhou, "Practical blacklist-based anonymous authentication scheme for mobile crowd sensing," *Peer-to-Peer Networking and Applications*, vol. 9, no. 4, pp. 1–12, 2015.

[45] J. Zhou, Z. Cao, and X. Dong, "Secure and efficient fine-grained multiple file sharing in cloud-assisted crowd sensing networks," *Peer-to-Peer Networking and Applications*, vol. 9, no. 4, pp. 1–21, 2016.

[46] J. Ni, Y. Yu, Y. Mu, and Q. Xia, "On the security of an efficient dynamic auditing protocol in cloud storage," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 10, pp. 2760–2761.

[47] F. Qiu, F. Wu, and G. Chen, "Privacy and quality preserving multimedia data aggregation for participatory sensing systems," *IEEE Transactions on Mobile Computing*, vol. 14, no. 6, pp. 1287–1300, 2015.

[48] J. Liu, H. Cao, Q. Li, F. Cai, X. Du, and M. Guizani, "A large-scale concurrent data anonymous batch verification scheme for mobile healthcare crowd sensing," *IEEE Internet of Things Journal*, to appear.

[49] T. Li, T. Jung, Z. Qiu, H. Li, L. Cao, and Y. Wang, "Scalable privacy-preserving participant selection for mobile crowdsensing systems: Participant grouping and secure group bidding," *IEEE Transactions on Network Science and Engineering*, to appear.

[50] L. Zhang, Y. Li, X. Xiao, X.-Y. Li, J. Wang, A. Zhou, and Q. Li, "Crowdbuy: Privacy-friendly image dataset purchasing via crowdsourcing," 2018.

[51] D. Wu, S. Si, S. Wu, and R. Wang, "Dynamic trust relationships aware data privacy protection in mobile crowd-sensing," *IEEE Internet of Things Journal*, to appear.

[52] J. Liu, F. Cai, L. Wu, R. Sun, L. Zhu, and X. Du, "Epda: Enhancing privacy-preserving data authentication for mobile crowd sensing," in *Proc. of GLOBECOM*, 2017, pp. 1–6.

[53] S. Rahaman, L. Cheng, D. D. Yao, H. Li, and J.-M. J. Park, "Provably secure anonymous-yet-accountable crowdsensing with scalable sublinear revocation," *Proceedings on Privacy Enhancing Technologies*, vol. 2017, no. 4, pp. 384–403, 2017.

[54] J. Ren, Y. Zhang, K. Zhang, and X. S. Shen, "Sacrm: social aware crowdsourcing with reputation management in mobile sensing," *Computer Communications*, vol. 65, pp. 55–65, 2015.

[55] L. Kazemi, C. Shahabi, and L. Chen, "Geotrucrowd: trustworthy query answering with spatial crowdsourcing," in *Proc. of ACM SIGSPATIAL GIS*, 2013, pp. 314–323.

[56] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proceeding of the VLDB Endowment*, vol. 7, no. 10, pp. 919–930, 2014.

[57] Y. Shen, L. Huang, L. Li, X. Lu, S. Wang, and W. Yang, "Towards preserving worker location privacy in spatial crowdsourcing," in *Proc. of GlobeCom*, 2015, pp. 1–6.

[58] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *ACM Transactions on Spatial Algorithms and Systems*, vol. 1, no. 1, article 2, 2015.

[59] H. Xiong, D. Zhang, L. Wang, J. P. Gibson, and J. Zhu, "Eemc: Enabling energy-efficient mobile crowdsensing with anonymous participants," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 3, article 39, 2015.

[60] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. Barnes, "icrowd: Near-optimal task allocation for piggyback crowdsensing," *IEEE Transactions on Mobile Computing*, to appear.

[61] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang, "Ccs-ta: quality-guaranteed online task allocation in compressive crowdsensing," in *Proc. of UbiComp*, 2015, pp. 683–694.

[62] C. H. Liu, B. Zhang, X. Su, J. Ma, W. Wang, and K. K. Leung, "Energy-aware participant selection for smartphone-enabled mobile crowd sensing," *IEEE Systems Journal*, to appear.

[63] M. Zhang, P. Yang, C. Tian, S. Tang, X. Gao, B. Wang, and F. Xiao, "Quality-aware sensing coverage in budget constrained mobile crowdsensing networks," *IEEE Transactions on Vehicular Technology*, to appear.

[64] I. B. Amor, S. Benbernou, M. Ouziri, Z. Malik, and B. Medjahed, "Discovering best teams for data leak-aware crowdsourcing in social networks," *ACM Transactions on the Web*, vol. 10, no. 1, article 2, 2016.

[65] T. Kandappu, V. Sivaraman, A. Friedman, and R. Boreli, "Loki: a privacy-conscious platform for crowdsourced surveys," in *Proc. of COMSNETS*, 2014, pp. 1–8.

[66] L. Pournajaf, D. A. Garcia-Ulloa, L. Xiong, and V. Sunderam, "Participant privacy in mobile crowd sensing task management: A survey of methods and challenges," *ACM SIGMOD Record*, vol. 44, no. 4, pp. 23–34, 2016.

[67] H. To, C. Shahabi, and L. Xiong, "Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server," in *Proc. of IDE*, 2018.

[68] L. Xiao, D. Jiang, D. Xu, and N. An, "Secure mobile crowdsensing with deep learning," *arXiv preprint arXiv:1801.07379*, 2018.

[69] J. Chen, H. Ma, D. Zhao, and L. Liu, "Correlated differential privacy protection for mobile crowdsensing," *IEEE Transactions on Big Data*, to appear.

[70] Y. Sei and A. Ohsuga, "Differential private data collection and analysis based on randomized multiple dummies for untrusted mobile crowdsensing," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 4, pp. 926–939, 2017.

[71] S. He, D.-H. Shin, J. Zhang, and J. Chen, "Toward optimal allocation of location dependent tasks in crowdsensing," in *Proc. of IEEE INFOCOM*, 2014, pp. 745–753.

[72] Y. Gong, L. Wei, Y. Guo, C. Zhang, and Y. Fang, "Optimal task recommendation for mobile crowdsourcing with privacy control," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 745–756, 2016.

[73] L. R. Varshney, A. Vempaty, and P. K. Varshney, "Assuring privacy and reliability in crowdsourcing with coding," in *Proc. of ITA*, 2014, pp. 1–6.

[74] H. Kajino, H. Arai, and H. Kashima, "Preserving worker privacy in crowdsourcing," *Data Mining and Knowledge Discovery*, vol. 28, no. 5-6, pp. 1314–1335, 2014.

[75] Y. He, L. Sun, Z. Li, H. Li, and X. Cheng, "An optimal privacy-preserving mechanism for crowdsourced traffic monitoring," in *Proc. of MobiHoc*, 2014, pp. 11–18.

[76] X. Chen, X. Wu, X.-Y. Li, Y. He, and Y. Liu, "Privacy-preserving high-quality map generation with participatory sensing," in *Proc. of IEEE INFOCOM*, 2014, pp. 2310–2318.

[77] S. Chang, H. Zhu, W. Zhang, L. Lu, and Y. Zhu, "Pure: Blind regression modeling for low quality data with participatory sensing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 4, pp. 1199–1211, 2016.

[78] J. Hu, H. Lin, X. Guo, and J. Yang, "Dtcs: An integrated strategy for enhancing data trustworthiness in mobile crowdsourcing," *IEEE Internet of Things Journal*, to appear.

[79] Ú. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proc. of ACM CCS*, 2014, pp. 1054–1067.

[80] S. Wang, L. Huang, M. Tian, W. Yang, H. Xu, and H. Guo, "Personalized privacy-preserving data aggregation for histogram estimation," in *Proc. of GlobeCom*, 2015, pp. 1–6.

[81] J. Chen, H. Ma, and D. Zhao, "Private data aggregation with integrity assurance and fault tolerance for mobile crowd-sensing," *Wireless Networks*, to appear.

[82] S. Wang, L. Huang, P. Wang, Y. Shen, H. Xu, and W. Yang, "Privacy preserving big histogram aggregation for spatial crowdsensing," in *Proc. of IPCCC*, 2015, pp. 1–8.

[83] J. Chen, H. Ma, D. S. Wei, and D. Zhao, "Participant-density-aware privacy-preserving aggregate statistics for mobile crowd-sensing," in *Proc. of ICPADS*, 2015, pp. 140–147.

[84] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Rescuedp: Real-time spatio-temporal crowd-sourced data publishing with differential privacy," in *Proc. of IEEE INFOCOM*, 2016, pp. 1–9.

[85] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proc. of SenSys*, 2015, pp. 183–196.

[86] J. Chen, H. Ma, and D. Zhao, "Private data aggregation with integrity assurance and fault tolerance for mobile crowd-sensing," *Wireless Networks*, vol. 23, no. 1, pp. 131–144, 2017.

[87] Y. Zhang, H. Zhang, S. Tang, and S. Zhong, "Designing secure and dependable mobile sensing mechanisms with revenue guarantees," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 1, pp. 100–113, 2016.

[88] J. Sun and H. Ma, "Privacy-preserving verifiable incentive mechanism for online crowdsourcing markets," in *Proc. of ICCCN*, 2014, pp. 1–8.

[89] T. Dimitriou and I. Krontiris, "Privacy-respecting auctions as incentive mechanisms in mobile crowd sensing," in *Proc. of WISTP*, 2015, pp. 20–35.

[90] J. Xu, J. Xiang, and D. Yang, "Incentive mechanisms for time window dependent tasks in mobile crowdsensing," *IEEE Transactions on Wireless Communications*, vol. 14, no. 11, pp. 6353–6364, 2015.

[91] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *Proc. of ICDCS*, 2016, pp. 344–353.

[92] X. Jin, R. Zhang, Y. Chen, T. Li, and Y. Zhang, "Dpsense: Differentially private crowdsourced spectrum sensing," in *Proc. of ACM CCS*, 2016, pp. 296–307.

[93] Y. Wang, Z. Cai, G. Yin, Y. Gao, X. Tong, and G. Wu, "An incentive mechanism with privacy protection in mobile crowdsourcing systems," *Computer Networks*, vol. 102, pp. 157–171, 2016.

[94] Q. Li and G. Cao, "Providing privacy-aware incentives in mobile sensing systems," *IEEE Transactions on Mobile Computing*, vol. 15, no. 6, pp. 1485–1498, 2016.

[95] X. Niu, M. Li, Q. Chen, Q. Cao, and H. Wang, "Eppi: An e-cent-based privacy-preserving incentive mechanism for participatory sensing systems," in *Proc. of IPCCC*, 2014, pp. 1–8.

[96] S. Delgado-Segura, C. Tanas, and J. Herrera-Joancomartí, "Reputation and reward: Two sides of the same bitcoin," *Sensors*, vol. 16, no. 6, p. 776, 2016.

[97] C. Tanas, S. Delgado-Segura, and J. Herrera-Joancomartí, "An integrated reward and reputation mechanism for mcs preserving users privacy," in *Proc. of DPM*, 2015, pp. 83–99.

[98] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, "Security, privacy & incentive provision for mobile crowd sensing systems," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 839–853, 2016.

[99] S. Gisdakis, P. Papadimitratos, and A. Giannetsos, "Data verification and privacy-respecting user remuneration in mobile crowd sensing," 2015.

[100] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems," in *Proc. of MobiHoc*, vol. 16, 2016, pp. 341–350.

[101] Y. Hui, Z. Su, and S. Guo, "Utility based data computing scheme to provide sensing service in internet of things," *IEEE Transactions on Emerging Topics in Computing*, 2017.

[102] X. Zhang, Z. Yang, Y. Liu, J. Li, and Z. Ming, "Toward efficient mechanisms for mobile crowdsensing," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, pp. 1760–1771, 2017.

[103] L. Xiao, T. Chen, C. Xie, H. Dai, and V. Poor, "Mobile crowdsensing games in vehicular networks," *IEEE Transactions on Vehicular Technology*, to appear.

[104] K. Yang, K. Zhang, J. Ren, and X. Shen, "Security and privacy in mobile crowdsourcing networks: challenges and opportunities," *IEEE communications magazine*, vol. 53, no. 8, pp. 75–81, 2015.

[105] I. Krontiris, M. Langheinrich, and K. Shilton, "Trust and privacy in mobile experience sharing: future challenges and avenues for research," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 50–55, 2014.

[106] J. Ni, A. Zhang, X. Lin, and X. S. Shen, "Security, privacy, and fairness in fog-based vehicular crowdsensing," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 146–152, 2017.

[107] A. Alamer, J. Ni, X. Lin, and X. Shen, "Location privacy-aware task recommendation for spatial crowdsourcing," in *Proc. of WCSP*, 2017, pp. 1–6.

[108] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *Proc. of MobiCom*, 2011, pp. 145–156.

[109] J. Ni, K. Zhang, K. Alharbi, X. Lin, N. Zhang, and X. S. Shen, "Differentially private smart metering with fault tolerance and range-based filtering," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2483–2493, 2017.

[110] J. Ni, K. Zhang, K. Alharbi, X. Lin, N. Zhang, and X. Shen, "Differentially private smart metering with fault tolerance and range-based filtering," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2483–2493, 2017.

[111] H. Mousa, S. B. Mokhtar, O. Hasan, O. Younes, M. Hadhoud, and L. Brunie, "Trust management and reputation systems in mobile participatory sensing applications: A survey," *Computer Networks*, vol. 90, pp. 49–73, 2015.

[112] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *Proc. of MDM*, 2014, pp. 73–82.

[113] J. Ni, K. Zhang, X. Lin, Q. Xia, and X. Shen, "Privacy-preserving mobile crowdsensing for located-based applications," in *Proc. of ICC*, 2017, pp. 1–6.

[114] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *arXiv preprint arXiv:1806.04057*, 2018.

[115] D. Cash, E. Kiltz, and V. Shoup, "The twin diffie-hellman problem and applications," in *Proc. of EUROCRYPT*, 2008, pp. 127–145.

[116] J. Ni, X. Lin, and X. Shen, "Efficient and secure service-oriented authentication supporting network slicing for 5g-enabled iot," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 644–657, 2018.

[117] Q. Wang, Y. Zhang, X. Lu, Z. Wang, Z. Qin, and K. Ren, "Real-time and spatio-temporal crowd-sourced social network data publishing with differential privacy," *IEEE Transactions on Dependable and Secure Computing*, to appear.

[118] H. Li, K. Ota, M. Dong, and M. Guo, "Mobile crowdsensing in software defined opportunistic networks," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 140–145, 2017.

[119] H. To, C. Shahabi, and L. Kazemi, "A server-assigned spatial crowdsourcing framework," *ACM Transactions on Spatial Algorithms and Systems*, vol. 1, no. 1, p. 2, 2015.

[120] X. Zhang, Z. Yang, W. Sun, Y. Liu, S. Tang, K. Xing, and X. Mao, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[121] M. Bellare, S. Keelveedhi, and T. Ristenpart, "Message-locked encryption and secure deduplication," in *Proc.of EUROCRYPT*, 2013, pp. 296–312.

[122] ——, "Dupless: Server-aided encryption for deduplicated storage." *Proc. Usenix Security*, vol. 2013, pp. 179–194.

[123] X. Yi, A. Bouguettaya, D. Georgakopoulos, A. Song, and J. Willemson, "Privacy protection for wireless medical sensor data," *IEEE transactions on dependable and secure computing*, vol. 13, no. 3, pp. 369–380, 2016.

[124] L. M. Vaquero and L. Rodero-Merino, "Finding your way in the fog: Towards a comprehensive definition of fog computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.

[125] J. Ni, K. Zhang, X. Lin, and X. Shen, "Securing fog computing for internet of things applications: Challenges and solutions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 1, pp. 601–628, 2018.

[126] J. Ni, K. Zhang, Y. Yu, X. Lin, and X. Shen, "Providing task allocation and secure deduplication for mobile crowdsensing via fog computing," *IEEE Transactions on Dependable and Secure Computing*, to appear.

[127] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in *Proc. of ASIACRYPT*, 2001, pp. 514–532.

[128] Q. Wu, Y. Mu, W. Susilo, B. Qin, and J. Domingo-Ferrer, "Asymmetric group key agreement," in *Proc.of EUROCRYPT*, 2009, pp. 153–170.

[129] A. Shamir and Y. Tauman, "Improved online/offline signature schemes," in *Proc. of CRYPTO*, 2001, pp. 355–367.

[130] S.-H. Chang and Z.-R. Chen, "Protecting mobile crowd sensing against sybil attacks using cloud based trust management system," *Mobile Information Systems*, vol. 2016, 2016.

[131] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, and A. Vecchio, "Smartphone-based crowdsourcing for network monitoring: Opportunities, challenges, and a case study," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 106–113, 2014.

[132] J. Camenisch and M. Stadler, "Efficient group signature schemes for large groups," in *Proc. of CRYPTO*, 1997, pp. 410–424.

[133] J. Scott, R. Gass, J. Crowcroft, P. Hui, C. Diot, and A. Chaintreau, "Crawdad trace cambridge/haggle/imote/infocom (v. 2006-01-31)," 2006.

[134] K. Zhang, X. Liang, R. Lu, K. Yang, and X. S. Shen, "Exploiting mobile social behaviors for sybil detection," in *Proc.of INFOCOM*, 2015, pp. 271–279.

[135] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart gridthe new and improved power grid: A survey," *IEEE communications surveys & tutorials*, vol. 14, no. 4, pp. 944–980, 2012.

[136] J. Ni, K. Zhang, X. Lin, and X. S. Shen, "Edat: Efficient data aggregation without ttp for privacy-assured smart metering," in *Communications (ICC), 2016 IEEE International Conference on*, 2016, pp. 1–6.

[137] J. Ni, K. Alharbi, X. Lin, and X. Shen, "Security-enhanced data aggregation against malicious gateways in smart grid," in *Global Communications Conference (GLOBECOM), 2015 IEEE*, 2015, pp. 1–6.

[138] X. Li, X. Liang, R. Lu, X. Shen, X. Lin, and H. Zhu, "Securing smart grid: cyber attacks, countermeasures, and challenges," *IEEE Communications Magazine*, vol. 50, no. 8, 2012.

[139] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building*. ACM, 2010, pp. 61–66.

[140] F. Cleveland, "Iec tc57 wg15: Iec 62351 security standards for the power system information infrastructure," *White Paper*, 2012.

[141] T. Dierks, "The transport layer security (tls) protocol version 1.2," 2008.

[142] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[143] H.-Y. Lin, W.-G. Tzeng, S.-T. Shen, and B.-S. P. Lin, "A practical smart metering system supporting privacy preserving billing and load monitoring," in *Proc. of ACNS*, 2012, pp. 544–560.

[144] C. Rottondi, G. Verticale, and A. Capone, "Privacy-preserving smart metering with multiple data consumers," *Computer Networks*, vol. 57, no. 7, pp. 1699–1713, 2013.

[145] T. Dimitriou and G. Karame, "Privacy-friendly tasking and trading of energy in smart grids," in *Proc. of ACM SAC*, 2013, pp. 652–659.

[146] C. Rottondi, M. Savi, G. Verticale, and C. Krauß, "Mitigation of p2p overlay attacks in the automatic metering infrastructure of smart grids," 2012.

[147] H. Krawczyk and T. Rabin, "Chameleon signatures." in *Proc. of NDSS*, 2000.

[148] H. Shacham and B. Waters, "Compact proofs of retrievability," in *Proc. of Asiacrypt*, 2008, pp. 90–107.

[149] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," in *Proc. of TCC*, 2005, pp. 325–341.

[150] D. Derler and D. Slamanig, "Key-homomorphic signatures and applications to multiparty signatures and non-interactive zero-knowledge," IACR Cryptology ePrint Archive, 2016: 792, Tech. Rep., 2016.

[151] J. M. Pollard, "Kangaroos, monopoly and discrete logarithms," *Journal of cryptology*, vol. 13, no. 4, pp. 437–447, 2000.

[152] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE transactions on information theory*, vol. 31, no. 4, pp. 469–472, 1985.

[153] B. Libert and J.-J. Quisquater, "Improved signcryption from q-diffie-hellman problems," in *Proc. of SCN*, 2004, pp. 220–234.

[154] C.-I. Fan, S.-Y. Huang, and Y.-L. Lai, "Privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Transactions on Industrial informatics*, vol. 10, no. 1, pp. 666–675, 2014.

[155] K. Ohara, Y. Sakai, F. Yoshida, M. Iwamoto, and K. Ohta, "Privacy-preserving smart metering with verifiability for both billing and energy management," in *Proc. of AsiaPKC*, 2014, pp. 23–32.

[156] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. of EUROCRYPT*, 1999, pp. 223–238.

[157] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "Greengps: a participatory sensing fuel-efficient maps application," in *Proc. of MobiSys*, 2010, pp. 151–164.

[158] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification," *ACM Transactions on Sensor Networks*, vol. 11, no. 4, article 55, 2015.

[159] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li, "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Proc. of Mobicom*, 2014, pp. 249–260.

[160] D. Yang, G. Xue, X. Fang, and J. Tang, "Incentive mechanisms for crowdsensing: Crowdsourcing with smartphones," *IEEE/ACM Transactions on Networking*, vol. 24, no. 3, pp. 1732–1744, 2016.

[161] H. Xie and J. C. Lui, "Incentive mechanism and rating system design for crowdsourcing systems: Analysis, tradeoffs and inference," *IEEE Transactions on Services Computing*, to appear.

[162] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "Trac: Truthful auction for location-aware collaborative sensing in mobile crowdsourcing," in *Proc. of IEEE INFOCOM*, 2014, pp. 1231–1239.

[163] Y. Wen, J. Shi, Q. Zhang, X. Tian, Z. Huang, H. Yu, Y. Cheng, and X. Shen, "Quality-driven auction-based incentive mechanism for mobile crowd sensing," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 9, pp. 4203–4214, 2015.

[164] D. Zhao, X.-Y. Li, and H. Ma, "Budget-feasible online incentive mechanisms for crowdsourcing tasks truthfully," *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 647–661, 2016.

[165] J. Ni, X. Lin, K. Zhang, and Y. Yu, "Secure and deduplicated spatial crowdsourcing: A fog-based approach," in *Proc. of IEEE Globecom*, 2016, pp. 1–6.

[166] J. Ni, X. Lin, Q. Xia, and X. Shen, "Dual-anonymous reward distribution for mobile crowdsensing," in *Proc. of ICC*, 2017, pp. 1–6.

[167] Y. Zhang, Y. Mao, H. Zhang, and S. Zhong, "Privacy preserving market schemes for mobile sensing," in *Parallel Processing (ICPP), 2015 44th International Conference on*, 2015, pp. 909–918.

[168] I. Evtimov, K. Eykholt, E. Fernandes, T. Kohno, B. Li, A. Prakash, A. Rahmati, and D. Song, "Robust physical-world attacks on deep learning models," *arXiv preprint arXiv:1707.08945*, 2017.

Appendix List of Publications

1. 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*, Under Review.

2. J. Ni, K. Zhang, Y. Yu, X. Lin, and X. Shen, "Providing Task Allocation and Secure Deduplication for Mobile Crowdsensing via Fog Computing", *IEEE Transactions on Dependable and Secure Computing*, DOI: 10.1109/TDSC.2018.2791432, Jan. 2018.

3. J. Ni, K. Zhang, X. Lin, and X. Shen, "Balancing Security and Efficiency for Smart Metering against Misbehaving Collectors", *IEEE Transactions on Smart Grid*, DOI: 10.1109/TSG.2017.2761804, Oct. 2017.

4. J. Ni, X. Lin, Q. Xia, and X. Shen, "Dual-Anonymous Reward Distribution for Mobile Crowdsensing", in Proc. of *IEEE ICC* – IEEE International Conference on Communications, Paris, France, May 21–25, 2017.

5. J. Ni, K. Zhang, X. Lin, Q. Xia, and X. Shen "Privacy-Preserving Mobile Crowdsensing for Location-Based Applications", in Proc. of *IEEE ICC* – IEEE International Conference on Communications, Paris, France, May 21–25, 2017.

6. J. Ni, X. Lin, K. Zhang, and Y. Yu, "Secure and Deduplicated Spatial Crowdsourcing: A Fog-Based Approach", in Proc. of IEEE *Globecom* – IEEE Global Communication Conference, Washington D.C., USA, Dec. 4–8, 2016.