Efficient and Privacy-Preserving Worker Selection in Mobile Crowdsensing

by

Xichen Zhang

Master of Computer Science, University of New Brunswick, Canada, 2018
Bachelor of Engineering, Changsha University of Science and Technology, China, 2010

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

In the Graduate Academic Unit of Computer Science

Supervisor(s):	Rongxing Lu, Ph.D., Faculty of Computer Science
	Ali A. Ghorbani, Ph.D., Faculty of Computer Science
Examining Board:	Suprio Ray, Ph.D., Faculty of Computer Science
	Saqib Hakak, Ph.D., Faculty of Computer Science
	Howard Li, Ph.D.,
	Department of Electrical and Computer Engineering
External Examiner:	Amiya Nayak, Ph.D., Faculty of Engineering,
	University of Ottawa

This dissertation is accepted by the
Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

May, 2022

Abstract

The rapid growth of the Internet of Things (IoTs) has enabled a new sensing paradigm, called Mobile Crowdsensing (MCS). A crowd of mobile participants, namely workers, are selected by the MCS platform to outsource their real-time sensing data for specific tasks, such as location recommendation, air quality monitoring, and traffic monitoring. Work selection is the process of allocating qualified workers to suitable MCS tasks. In MCS, workers' reliability and their sensing data quality play significant roles in the service quality. Therefore, worker selection is always one of the most fundamental problems in MCS applications.

Despite the considerable potential and extensive development, there are still several challenges and issues in MCS services which cannot be ignored. i) In worker selection, it is inevitable for the workers to share some of their personal sensitive information, which may be exploited by hostile MCS platform for malicious activities. Therefore, privacy preservation for workers' sensitive information is a major concern in MCS platforms. ii) Evaluating workers' trustability or credibility is one of the most essential issues that the MCS platform needs to solve. But these attributes were often neglected in previous literature. iii) Worker selection is a dynamic process where the workers can continuously arrive at/leave the platform. However, most of the existing studies only focus on selecting workers statically. iv) In MCS tasks, the workers are always heterogeneous with different computational resources. Therefore, the MCS platform needs to select qualified workers in terms of their various comput-

ing characteristics. v) Worker's real-time spatial-temporal information plays a vital role in worker selection, and should be paid more attention in designing real-world MCS applications.

In this thesis, we focus on efficient and privacy-preserving worker selection in MCS and propose several schemes for addressing the above challenges. Specifically, the main contributions are as follows. i) We proposed a privacy-preserving worker selection scheme based on probabilistic skyline computation for calculating workers' trustability based on their historical reviews. ii) We proposed a privacy-preserving dynamic worker selection scheme based on probabilistic skyline query over sliding windows, which can continuously and dynamically select qualified workers. iii) We proposed a novel application scenario called Federated MCS by integrating the concept of Federated Learning with MCS. The proposed scheme can select qualified workers based on the group skyline technique and aggregate local model updates for training the global model. iv) We conducted a series of studies on worker selection regarding spatial-temporal matching and evaluation. The proposed privacy-preserving schemes can select qualified workers in terms of their real-time spatial-temporal information.

The research findings and experimental results in this thesis should be useful for selecting qualified workers effectively and securely in MCS applications.

Dedication

To my beloved families and respectful teachers without whom I would have achieved nothing.

Acknowledgements

I would like to thank all the people who greatly support me to finish my Ph.D. study in the past four years at the University of New Brunswick, Canadian Institute for Cybersecurity. They are my supervisors, my families, my thesis committee members, my colleagues, and my friends. Without your continuous help, support, encouragement, and guidance, I could not make it possible.

First and foremost, I would like to express my deepest gratitude to my supervisors, Dr. Rongxing Lu and Dr. Ali A. Ghorbani, for their invaluable advice, unwavering support, consistent help, and informative guidance throughout this thesis. Dr. Lu assists and supervises me at every stage of my Ph.D. thesis. His immense knowledge and plentiful experience create fertile grounds for me to learn new stuff and improve myself during my study. I am grateful for his helpful assistance, insightful suggestions, and forward-looking guidance. Dr. Lu not only teaches me how to become a good Ph.D. student, but also tells me how to become a qualified researcher in the future. In addition, Dr. Ghorbani creates an excellent atmosphere for me to produce innovative ideas and work plans during my Ph.D. study. He always trusts young researchers and students and give them opportunities and platforms to show their self-worth. I appreciate that Dr. Ghorbani provides me a chance to join the research group as a Master's student in 2015. Since then, his insightful advice and powerful encouragement motivate me to generate confidence in doing research and widen my view from various perspectives. In a nutshell, my supervisors, upon whose shoulder I stand, explored and paved the path before me. Without them, this thesis is not even possible. I could not have imagined having better supervisors like Dr. Lu and Dr. Ghorbani.

Besides my supervisors, I would like to thank my thesis advisory committee members, Dr. Suprio Ray and Dr. Saqib Hakak, my thesis external examiner, Dr. Amiya Nayak, for their time and efforts in patiently reading my thesis and their invaluable advice and recommendations on how to improve my thesis. Especially, I would like to express my sincere gratitude to Dr. Suprio, for his patient and precious supervision for my reading course, which enlightens me at the first glance of a new research direction. I really enjoy the weekly discussions with him during the reading course program, and I am looking forward to the open opportunities for us to do research again in the future.

I would also like to express my sincere appreciation to Dr. Wei Song, Dr. Jun Shao and Dr. Hui Zhu, for the helpful suggestions, insightful comments, and careful revisions on my publications and this thesis.

In addition, I would like to offer my special thanks to my fellow in the University of New Brunswick, Canadian Institute for Cybersecurity: Ms. Yandong Zheng, Mr. Songnian Zhang, Mr. Yunguo Guan, Dr. Hassan Mahdikhani, Dr. Samaneh Mahdavifar, Mr. Fan Yin, Dr. Xue Yang, Dr. Beibei Li, Dr. Fengwei Wang, Mr. Xiapeng Yang, Dr. Xiong Li, Dr. Yonggang Zhang, Dr. Dongyang Xu, Dr. Arash Habibi Lashkari, Dr. Farzane Shoaleh, Ms. Ruth Murgatroyd, Dr. Haruna Isah, Dr. Sajjad Dadkhah, Ms. Pamela Kitchen, Mr. Faruk Ener, Mr. Mojtaba Nadali, and Mr. Mohamed Salem, for all your support and help in various perspectives. There are many other people whose names are not presented here, but they hold a special place in my heart.

Finally, and most importantly, I would like to express my heartfelt thanks to my parents, my wife, and my son. Thanks to my parents for consistently supporting me and understanding me in any way possible. You are the motivation and direction of my life. Thanks to my wife for tolerating me in any way possible. You are the driving force in my life. Thanks to my son for bringing me so much happiness and hope. You are the sunshine in my life. My special thanks go to my wife, Jiejie Wang, who is a great partner and also an excellent researcher, for the loving support and understanding for me to complete my Ph.D. study.

Table of Contents

List of Tables

List of Figures

Abbreviations

AMT	Amazon Mechanical Turks. 3
BF	Bloom Filter. 27
C-skyline	Constrained Skyline. 21
CG-skyline	Constrained Group Skyline. 22
FPMT	Few Participants with More Tasks. 15
FL	Federated Learning. 97
F-MCS	Federated Mobile Crowdsensing. 97
G-skyline	Group Skyline. 21
HE	Homomorphic Encryption. 98
IoT	Internet of Things. 1
MCS	Mobile Crowdsensing. ii, 1
MPFT	More Participants with Few Tasks. 15
NIEC	Non-interactive Encrypted Integer Comparison Protocol. 33
PKE	Public Key Encryption. 24
PS-score	Probabilistic Skyline Score. 31
SHE	Symmetric Homomorphic Encryption. 25

Chapter 1

Introduction

1.1 Worker Selection in MCS

Recently, the Internet of Things (IoTs) underwent extensive development and has been deemed one of the most significant drivers for information and industrial revolution [134, 130]. It is predicted that the number of smart devices will be 500 billion by 2030 [13]. With a large number of smart devices connected to the Internet, IoTs creates fertile grounds for sowing novel applications and services, including smart home, smart traffic, smart city, smart manufacturing, cloud computing, federated learning, and so on. Consequently, the explosion in the availability of smart devices brings a new paradigm of sensing network, called Mobile Crowdsensing (MCS), which has recently spurred lots of interests in both industries and academia [130, 134, 135]. By exploiting sensors embedded in mobile devices, MCS can be used in a large variety of real-world applications, such as location recommendation [101], air quality and environmental monitoring [79], traffic information sharing [128], traffic management [20, 109], health care provisioning [87], location-based social recommendation [88, 62], and point-of-interest characterization [22]. With the pervasive sensor-embedded smart devices, the MCS platform can utilize the dynamics and mobility of numerous workers and leverage large-scale and long-term crowdsensing services [136, 111, 123].

As a representative example, *WAZE* is a GPS navigation software owned by *Google*, which can provide real-time services like traffic monitoring and route recommendation with users' smartphones and tablets [113]. The users can get the best routes with real-time alerts for traffic, police, accidents, and more. *TripAdvisor* is another typical example of a social recommendation application based on MCS. As one of the world's largest travelling recommendation platforms, *TripAdvisor* provides millions of real-time reviews about restaurants, experiences, things to do, places of interest, airlines, and cruises with user-generated content.

MCS requires a large number of participants (i.e., workers) to sense the surrounding environment using the sensing devices with built-in sensors [62]. In MCS services, worker selection is the process of allocating the sensing tasks to the proper workers. It has been regarded as one of the most fundamental issues in MCS, since the reliability of the workers and the trustability of their sensing data play significant roles in the service quality [135]. Recently, there has been extensive research on studying worker selection in MCS services [35, 45, 107, 111, 44, 34]. However, most of them evaluate workers based on the information provided by the workers. Such studies are infeasible for determining workers' real qualifications if there are strategic and selfish workers who try to maximize their profits by providing inaccurate information. In real-world MCS applications, workers' sensing performance is always task-by-task due to the limitations of sensing equipment and the dynamics of sensing conditions. Therefore, it is significant to measure workers' trustability and reliability based on their previous sensing behaviors.

In addition, to achieve a satisfactory result for worker selection, it is inevitable for the participating workers to share some of their sensitive information, e.g., trustability, working experience, real-time location. In particular, a hostile MCS platform may exploit workers' trustability for unfair competition [45]; an attacker may infer workers' daily behaviour by analysing their working experience of different MCS tasks. As a result, workers may be reluctant to take part in the MCS tasks due to the potential privacy disclosure [58]. To stimulate workers' interests and enthusiasm, an ideal MCS platform should protect workers' sensitive information while selecting proper workers efficiently and effectively.

1.2 The Challenges of Worker Selection in MCS

As we mentioned, each worker is the basic computational unit for the MCS task. Therefore, the selection of suitable and trustable workers is imperative for the MCS service quality. As worker selection is the core issue, in this section, we discuss some critical challenges for worker selection that may limit the extensive adoption of MCS applications.

- Privacy-preservation in worker selection. To select suitable and trustable workers for MCS tasks, it is inevitable for the participating workers to share some of their sensitive information, e.g., real-time spatial-temporal information, daily routine, working experience. In particular, a hostile MCS platform may exploit workers' trustability for unfair competition. In addition, an attacker can get access to workers' personal information by analysing their sensing data and history. Consequently, workers may be reluctant to take part in the tasks due to the potential privacy disclosure. To stimulate workers' motivation, an ideal MCS platform should consider privacy-preservation as a fundamental and essential component in the process of worker selection.

- The evaluation of user's trustability. The trustability of each participating worker is the key factor for a successful MCS service. In traditional crowd-sensing applications, the service provider can guarantee to select high-quality workers by different approaches. For example, the platform can provide training and testing processes for all the participating workers and only select the qualified workers. However, those approaches are not feasible for MCS, since the MCS tasks are always time-limited and the workers should be selected as fast as possible (e.g., select best taxi drivers for a customer). Some existing studies evaluate workers' qualifications based on the information provided by the workers. However, such approaches are not practical since the workers are strategic and selfish and try to maximize their own gains by providing misleading or wrong information. Essentially, evaluating workers' trustability or credibility is one of the most important issues that the MCS platform needs to solve.

- Dynamic worker selection in MCS. Most of the existing studies focus on selecting workers statically. Nevertheless, in many real-world MCS applications, an MCS task is a dynamic process and should be kept for a long-time period. Unlike the static scenario, such MCS tasks require a dynamic worker selection algorithm such that continuity and time-relevancy should be considered. Therefore, the MCS platform is more interested in recent information rather than those in the far past.

- Heterogeneous worker selection with limited computation resources. In MCS, the workers are always heterogeneous with different computational powers and resources. For instance, their mobile devices are different in terms of hardware, network connection, power status, CPU, and so on. Therefore, the workers with poor computational capacities will delay the whole sensing procedure and further degrade user experiences. As a result, the MCS platform needs to select suitable workers regarding their heterogeneous natures.

- Worker selection in terms of spatial-temporal factors. The spatial-temporal information matching between the MCS task and the workers is another fundamental problem for worker selection in MCS. Notably, the MCS platform needs to correctly calculate the spatial-temporal similarity between workers and the task and only select the workers with high spatial-temporal similarities. For example, an environment protection organization wants to hire mobile workers to monitor the real-time air pollution situation in a certain area during a specific time duration. The failure of selecting spatial-temporally matched workers may lead to serious consequences. For instance, the collected sensing data cannot reflect the underlying requirements of the task at all. Hence, spatial-temporal similarity matching should be paid more attention by both academia and industry in this domain.

1.3 Contributions of the Thesis

In this thesis, we devote our attention towards the problem of efficient and privacy-preserving worker selection in MCS and design several schemes for addressing the above challenges. The major contributions of this thesis are summarized as follows.

- To address the problem of evaluating workers' trustability, we have proposed a privacy-preserving worker selection scheme for MCS applications based on (probabilistic) skyline computation. Our proposed scheme can efficiently select suitable and reliable workers without revealing workers' personal information. Specifically, the privacy-preserving probabilistic skyline is used to calculate workers' *PS-score* based on historical reviews. *PS-score* reveals the fluctuation of workers' past performance and can be used as their trustability. Then, skyline computation is used for selecting workers in terms of ask price, task similarity, and trustability. A non-interactive encrypted integer comparison protocol called *NIEC* is designed for supporting our scheme, which can securely compare two integers in a certain range without revealing their real values.

- To address the problem of dynamic worker selection in MCS services, we have proposed a privacy-preserving worker selection scheme for MCS based on probabilistic skyline query over sliding windows. The proposed scheme can continuously select reliable workers in terms of working experience, expiry time, and trustability without revealing sensitive information. More specifically, a probabilistic skyline approach is designed for a dynamic situation where workers may constantly arrive at/leave the platform. For protecting workers' privacy, we have designed an ElGamal-based encryption approach for securely outsourcing sensitive information and comparing workers' personal information.

- To address the problem of heterogeneous worker selection in MCS services, we have proposed a privacy-preserving scheme for F-MCS applications, called FedSky. By extending the classic FedAvg algorithm, FedSky selects qualified workers based on CG-skyline technique and securely aggregates model updates for training the global model. Particularly, compared with FedAvg, our scheme considers the workers' dynamics and heterogeneous nature. It can significantly improve the efficiency of the model training process in F-MCS. Besides, we have designed a novel privacy-preserving data aggregation protocol based on the additive HE system. This protocol is exclusively designed for *cross-device* FL setting and requires no interactions between workers during the model training process.

- To address the problem of spatial-temporal worker selection in MCS services, we have proposed a series of privacy-preserving worker selection schemes in terms of both spatial-temporal similarity and spatial-temporal constraints. We introduced the definition and calculation of spatial-temporal similarity and constraints between participating workers and the MCS task in terms of intersection area ratio and overlapping time duration. For the case of spatial-temporal similarity, we designed a privacy-preserving approach for protecting participants' sensitive information based on the modified Paillier encryption technique. For the case of spatial-temporal constraint, we proposed a privacy-preserving worker selection scheme based on the symmetric homomorphic encryption (SHE) approach, which can protect participants' spatial information from being disclosed while still enabling effective worker assignment.

1.4 Outline of the Thesis

The remainder of the thesis is structured as follows. In chapter 2, we discuss some basic background information about MCS applications and services. In chapter 3, we review some fundamentals and background knowledge in this thesis. In chapter 4, we propose a privacy-preserving worker selection scheme based on probabilistic skyline computation. In specific, we calculate workers' trustability score with the probabilistic skyline value based on historical reviews. In chapter 5, we introduce a privacy-preserving and dynamic worker selection scheme based on probabilistic skyline query over sliding windows. The designed framework can continuously select reliable workers in terms of working experience, expiry time, and trustability without revealing sensitive information. In chapter 6, we introduce a novel sensing approach, called Federated MCS (F-MCS), which combines the concept of Federated Learning with MCS. In addition, we devise a privacy-preserving scheme called FedSky by extending the classic FedAvg algorithm. Our scheme can select qualified workers and securely aggregate model updates for training a global model by considering workers' dynamics and heterogeneous natures. Next, we focus on privacy-preserving worker selection in MCS in terms of spatial-temporal information. In particular, we study the problems of privacy-preserving worker selection regarding spatial-temporal similarity, spatial-temporal constraints, and workers' future spatial-temporal trajectory in chapter 7, 8, and 9, respectively. Finally, we recap the major contributions of the thesis and propose some interesting future directions in chapter 10.

Chapter 2

Background Information about MCS

In this chapter, we mainly discuss the background information about MCS, including the overview architecture of MCS, different application scenarios, and the major challenges and issues in MCS. After that, we present the state-of-the-art related work on worker selection and privacy-preservation in MCS.

2.1 The Overall Architecture of MCS Platform

Crowdsensing is the process of jointly organizing the crowd (i.e., workers) for solving human-intrinsic tasks [103]. Traditional crowdsensing applications are web-based platforms. For example, Wikipedia[1] is a crowdsensing-based online encyclopedia maintained by a community of online volunteers for editing, sharing, and delivering collaborative knowledge. Amazon Mechanical Turks (AMT)[2] is a crowdsensing website that remotely hires workers (they call turkers) to conduct on-demand and general-purpose tasks that computers are unable to accomplish, such as image

[1] https://www.wikipedia.org/
[2] https://www.mturk.com/

classification, sentiment analysis, emotion detection, audio transcription, and data collection and annotation. ImageNet[3] is a large visual database for visual object recognition based on crowdsensing. More than 14 million images are being hand-annotated by volunteers to indicate what objects are pictured. There are two types of data annotations in the ImageNet dataset. The image-level annotations indicate the presence or absence of an object class in an image. The object-level annotations provide a bounding box of each object in the image.

Recently, with the proliferation of smart devices and IoTs technologies, crowdsensing applications are shifting from the traditional web-based platforms to mobile platforms, i.e., MCS. Specifically, MCS is characterized by the following features [103].

- Each worker's credibility or trustability is crucial for MCS task quality. A more trustable worker should be more preferred, since he/she may have more working experience in similar tasks and his/her sensing data is more reliable in terms of accuracy, effectiveness and completeness.

- Each worker in MCS is considered as a mobile computing unit to complete tasks using their smart devices. Therefore, the computational powers of their smart devices are vital for service quality. The poor computational capacities can limit the overall performance of the sensing task.

- In most cases, each worker should physically move to the task location to accomplish the task. Consequently, the spatial information such as workers' current locations, mobility, and the associated contexts (e.g., the environment, the speed) are essential and should be considered for selecting suitable workers.

- The MCS task is usually real-time and time-limited, which requires the sensing platform to hire workers dynamically in terms of workers' temporal relevance.

[3] https://www.image-net.org/

Rather than statically selecting workers, the MCS platform should select workers regarding their arrival time, time duration for completing the task, and expiration time of the sensing data, etc. As a result, the temporal factor plays a disproportionate role in MCS for worker selection.

Particularly, an MCS platform contains three layers, i.e., the sensing layer, the communication layer, and the application layer. The overall architecture of the MCS platform is shown in Fig. 2.1. The details of each layer in the overall architecture are discussed as follows.

- **The Sensing Layer**: In this layer, the select workers should physically move to the task location and collect real-time sensing data with the built-in sensors in their smart devices. The sensors include an accelerometer, light sensor, GPS devices, radiation sensor, air quality sensor, health-care sensor, pressure sensor, noisy sensor, temperature sensor, and so on. The sensing layer is the most fundamental part of the MCS services. The workers' working experience, the power of their computational devices, and the reliability of the sensing data play significant roles in the service quality. Worker selection (i.e., task assignment), quality control, incentive mechanism, and privacy protection of workers' personal information are the major research problems for this layer [103].

- **The Communication Layer**: This layer comprises both technologies and methodologies about delivering sensing data from workers to the central platform [12]. The sensing data and other information can be communicated between workers and the MCS service provider by the base station, blue-tooth, WiFi, and network. The communication costs in this layer have drawn considerable attention in recent years, specifically after the extensive growth of federated learning-based sensing approaches. The information flow between

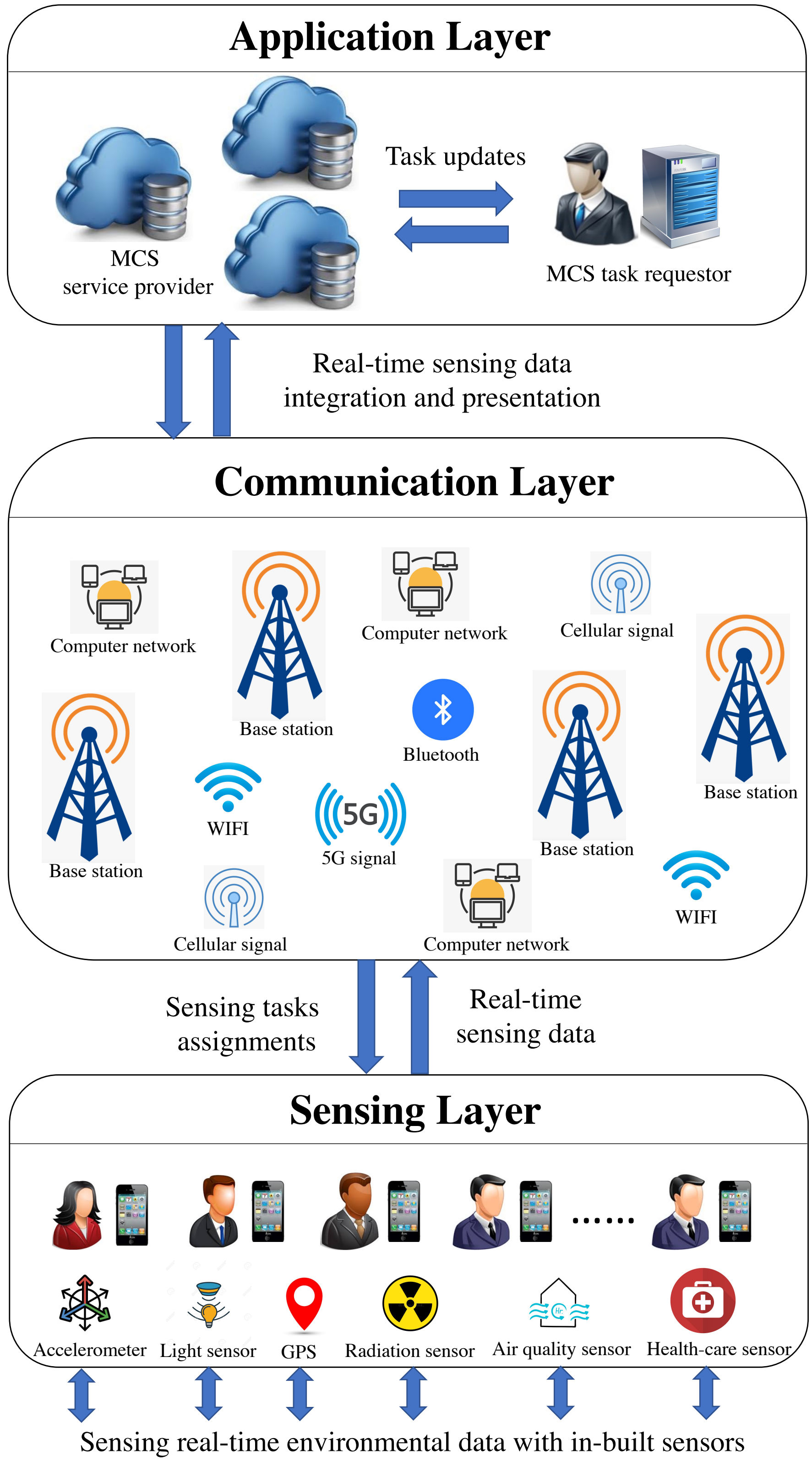


Figure 2.1: The overall three-layer architecture of the MCS platform.

this layer and the sensing layer includes (1) the MCS service provider assigns suitable workers for certain tasks and (2) the workers upload their real-time sensing data back to the central server.

- **The Application Layer**: This layer defines the high-level aspects of each MCS task, including the task content initialized by the task requestor, the specific requirements for the task, the expiration time for the task, the task location, and so on. The communication and information flow is mainly between the MCS service provider and the MCS task requestor. Before sending the sensing results to the requestor, the MCS service provider will usually integrate the sensing data from a large number of participating workers. Therefore, error data identification and duplicate data detection are major tasks for this layer.

2.2 The Domains of Interest in MCS Tasks

The MCS service has spurred lots of tasks in real-world application scenarios. Based on [9], there are major four categories of domains of interest for MCS tasks, they are environmental-related task, infrastructure-based task, social-based task, and behaviour-based task. Authors in [12] proposed a more fine-grained categorization of domains of interest. In their summary, the MCS task can be defined into the following aspects: emergency prevention and management, environmental monitoring, e-commerce, health-care, indoor localization, intelligent transportation system, mobile social network, public safety, unmanned vehicles urban planning, waste management, WiFi characterization, and others. Based on the previous literature, we proposed a hierarchy-based MCS task category in terms of the following domains of interest, they are (1) environmental-based task, (2) infrastructure-based task, (3)

Table 2.1: The hierarchy-based MCS task category in terms of domains of interest based on [62, 103, 12, 9].

Domains of Interests	Subcategory	Description	Reference
Environment task	Environmental monitoring	This task category provides real-time monitoring and managing of environmental conditions and resources, e.g., air quality, noisy level, pollution, and radiation.	[31, 40, 98, 27, 82]
	Waste management	This task category monitors and supports water-recycling operations, e.g., checking the amount of trash or daily waste collection routine.	[31, 7]
Infra. task	Emergency prevention and management	This task category can provide real-time evaluation and prevention of daily urban emergencies, e.g., the water level in the river bed, or the post-disaster reconstruction and services. The disaster can be earthquake, flood, fire, and so on.	[50, 89, 24, 29, 43]
	Indoor localization	This task category can help users to verify and navigate indoor locations.	[120, 121]
	WiFi characterization	This task category can map of WiFi coverage with different MCS techniques, e.g., exploiting passive interference power, measuring spectrum, and received power intensity.	[36]
	Intelligent transportation system	This task category can provide drivers real-time information about the traffic, including traffic jam, construction, accident. It can also plan the shortest path for the users. It can also monitor citizen mobility.	[108, 109, 55, 72, 146]
	Unmanned vehicles	This task category can support interaction between mobile users and vehicles for real-time information sharing and communication.	[126, 5, 59, 2, 65]
Daily task	E-commerce	In this task category, the users can collect, delivery, and share real-time e-commerce information.	[11, 26, 94, 95, 49]
	Mobile social networks	This task category can build social connections and relationship for users who have similar interests or habits.	[88, 17, 18]
	Public safety	This task category can help the government and citizens check the level of crimes for specific area or community.	[4, 141]
	Taxi calling and ridesharing	This task category can match the best drivers or taxis to customers by optimization algorithms.	[66, 129]
	Inventory and service identification	This task category can help users to check and identify the availability of the inventory or services.	[104, 73]
	Real-time data collection	This task category can help users collect real-time data based on their own requirements. For example, the MCS workers can collect diverse and real-time image data for a local landmark for 3D visual information construction.	[115, 114, 37]
	Service evaluation and recommendation	This task category can evaluate service qualities and provide real-time recommendation for the users.	[99, 74]
Health task	Health-care and well-being	This task can provide comprehensive health-care information and feedback by sharing and updating users' real-time physical or mental conditions.	[52, 1]
	Real-time heath-care monitoring	This task can provide real-time monitoring and evaluation for patients' health-care condition.	[71, 97]

daily-life task, and (4) health-care task. The details of each category and its corresponding sub-categories are summarized in Table 2.1.

2.3 The Related Research Problems in MCS Worker Selection

Recently, there are numerous endeavors that aim to study the problem of worker selection in MCS platforms. Here, we mainly review some related works in terms of the primary techniques, including privacy-preserving worker selection, incentive mechanism worker selection, reputation evaluation in worker selection, and task optimization in worker selection.

2.3.1 Privacy-preservation in MCS Worker Selection

As an emerging issue in MCS, recently, more studies focus on the problem of privacy-enhanced worker selection. Ni et al. [75] described a privacy-preserving MCS framework for location-based applications, which can balance the trade-off between privacy preservation and task allocation. A matrix-based location matching mechanism is used by the service provider to achieve location-based task allocation without disclosing workers' sensing location. Proxy re-encryption technology is developed to enhance privacy preservation during the MCS services. Wang et al. [111] provided a personalized privacy-preserving task allocation framework for MCS applications. Each worker uploads the obfuscated distances and personal privacy level to the server instead of its true locations or distances to tasks. Particularly, they proposed a Probabilistic Winner Selection Mechanism to minimize the total travel distance with the obfuscated information from workers, by allocating each task to the worker who has the largest probability of being closest to it. Wang et al. [107] proposed a location privacy-preserving worker selection framework with geo-obfuscation for protecting workers' locations during task allocation. More specifically, by obfuscating their real locations under the guarantee of differential privacy, workers can protect their location privacy regardless of adversaries' prior knowledge. In addition, a mixed-integer non-linear programming problem is defined for minimizing workers' expected travel distance and optimizing worker assignments. Zhao et al. [139] designed a privacy-preserving and data quality-aware incentive scheme, called PACE. In specific, they first proposed a zero-knowledge model of data reliability estimation that can protect data privacy while assessing data reliability. Then, they quantified the data quality based on the deviation between reliable data and the ground truth. Finally, they distributed monetary rewards to task participants according to their data quality.

2.3.2 Incentive Mechanism in MCS Worker Selection

To inspire workers' motivations and interests in participating in the MCS services, an effective incentive mechanism is indispensable. Jin et al. [45] present an incentive mechanism for worker selection in MCS services, which can choose workers that are more likely to provide reliable data, while protecting workers' privacy as well. Particularly, they proposed INCEPTION, a novel MCS system framework that integrates an incentive, a data aggregation, and a data perturbation mechanism. It can select workers who are more likely to provide reliable data and compensates their costs for both sensing and privacy leakage. They validated the desirable properties of INCEPTION through theoretical analyses as well as extensive simulations. Gong et al. [32] devised truthful crowdsensing mechanisms for incentivizing strategic workers to truthfully reveal their private quality and truthfully make efforts as desired by the requester. In their work, the authors assumed that a strategic worker with low quality may pretend to have a high quality to receive a high reward from the requester. Under the proposed mechanisms, they showed that the requester can assign the task only to the best workers that have the smallest virtual valuation.

Wu et al. [116] proposed a context-aware multi-armed bandit incentive method for selecting high-quality workers in MCS systems. In their work, workers' service quality is evaluated by their context and cost. Then, by accurately assessing workers' quality information, a modified Thompson sampling approach is utilized for selecting reliable workers based on reinforcement learning. Jin et al. [44] proposed a differentially private incentive mechanism that preserves the privacy of each worker's bid, which is based on the single-minded reverse combinatorial auction. Specifically, they devised a differentially private, approximately truthful, individually rational, and computationally efficient mechanism that approximately minimizes the platform's total payment with a guaranteed approximation ratio.

2.3.3 Reputation Evaluation in MCS Worker Selection

In this domain of interest, the MCS platform wants to evaluate the reputation or credibility of each participant for worker selection, and the trustability of the sensing data as well. Zhang et al. [136] proposed a probabilistic skyline based worker selection method, which solved a fundamental problem of calculating workers' trustability based on their historical reviews. In addition, they designed a non-interactive encrypted integer comparison protocol to compare the skyline dominance relationship between workers securely. Skyline operator is used for selecting workers based on ask price and task relativity. This is the first work using a skyline-based method for worker selection in MCS platforms, which can consider the trade-off between multiple aspects and select workers that are not dominated by others. Ren et al. [90] introduced a socially aware reputation management scheme for selecting well-suited participants and allocating the task rewards in MCS services. Concretely, social attributes, task delays, and reputation are focused on under a fixed task budget. Moreover, a rewarding scheme is devised to measure the quality of the sensing reports and allocate reliable workers to certain MCS tasks under the consideration of the trustworthiness and cost performance of task participants. Sun et al. [100] studied the problem of truth discovery in a crowdsensing question answering system based on a contract-based privacy-preserving incentive mechanism, called PINTION. This framework can provide personalized payments for workers with different privacy demands as a compensation for privacy cost, while ensuring accurate truth discovery. The basic idea is that each worker chooses to sign a contract to specify a privacy-preserving level and a payment. Then the workers can submit their perturbed answers with the privacy-preserving level in return for that payment. Finally, the authors introduced a set of optimal contracts under both complete and incomplete information models. These contracts could maximize the truth discovery accuracy, while satisfying the budget feasibility, individual rationality, and incentive compatibility properties.

2.3.4 Task Optimization in MCS Worker Selection

Overall speaking, task optimization in worker selection is the procedure to assign the most suitable workers for each task in terms of different impacting factors, e.g., the overall budget, the spatial-temporal factors, the rewards for each worker, and the travelling distance between the worker and the task. Guo et al. [35] designed a MCS scheme for multi-task worker selection, which considers both workers' intentional movement for time-sensitive tasks and unintentional movement for delay-tolerant tasks. For time-sensitive tasks, workers are required to move to the task venue intentionally and the goal is to minimize the total distance moved. For delay-tolerant tasks, the proposed framework can select workers whose route is predicted to pass by the task venues, and the goal is to minimize the total number of workers. Kazemi and Shahabi [48] studied the problem of spatial task assignment for spatial crowdsensing. In their proposed taxonomy, the workers send their real-time locations to a centralized server and thereafter the server assigns to every worker his/her nearby tasks intending to maximize the overall number of assigned tasks. In this work, the authors formally define the maximum task assignment problem and identify its challenges. To maximize the overall number of assigned tasks, their scheme highly depends on the locations of mobile workers, in which workers' essential personal information might be disclosed. Xiong et al. [118] investigated an energy-efficient mobile crowdsensing framework, where task assignments and sensing results are transferred in parallel with phone calls. The proposed framework can guarantee that a minimum number of anonymous participants return sensor results within a specified time frame, while minimizing the waste of energy due to redundant task assignments and considering privacy concerns of participants as well. Evaluation is conducted based on a large-scale real-world phone call dataset, and the results show that their proposed framework outperforms baseline methods in terms of reducing overall energy consumption significantly. Liu et al. [63] studied the problem of multi-task allocation in MCS services. In particular, two typical multi-task MCS environments are considered in their work, e.g., FPMT (few participants with more tasks) and MPFT (more participants with few tasks). Unique mechanisms are devised for different scenarios with different optimization goals. Particularly, for FPMT, each worker is required to complete multiple tasks and the optimization goal is to maximize the total number of accomplished tasks while minimizing the total movement distance. For MPFT, each worker is selected to perform one task based on pre-registered working areas, and the optimization objective is to minimize total incentive payments while minimizing the total travelling distance.

Chapter 3

Technical Fundamentals

In this chapter, we introduce some fundamental theories and techniques, which serve as the principle components in this thesis. These basic theories and techniques include different variants of skyline queries and some classic cryptography techniques,

3.1 Different Variants of Skyline Queries

3.1.1 Skyline

Skyline computation is a well-known approach for multi-dimensional decision analysis [8] [145]. Given a dataset $\mathcal{D} = \{\gamma_1, \gamma_2, \gamma_3, ..., \gamma_D\}$ that contains D objects. Each object is in d dimensional space, where $\gamma_i = (\gamma[1], \gamma[2], \gamma[3], ..., \gamma[d])$ for $i \in [1, D]$. For simiplicity, we assume for each dimension, larger values are more preferred. Let γ_a and γ_b denote two different objects in $\mathcal{D}$ where $a, b \in [1, D]$ and $a \neq b$. We define γ_a dominates γ_b, denoted as $\gamma_a \prec \gamma_b$, if for all $k \in [1, d]$, $\gamma_a[k] \geq \gamma_b[k]$, and there exists at least one k such that $\gamma_a[k] > \gamma_b[k]$. The skyline points of $\mathcal{D}$ are all the objects that are not dominated by others in $\mathcal{D}$. If $\gamma_a \not\prec \gamma_b$, it means either $\gamma_b \prec \gamma_a$ or they do not dominate each other. Notably, $\gamma_a \not\prec \gamma_a$ (i.e., γ_a does not dominate itself) because we have $\gamma_a[k] = \gamma_a[k]$ for $k \in [1, d]$.

Table 3.1: The dominance relation between objects A and B

	a_1	a_2	a_3	a_4
b_1	$a_1 \prec b_1$	NA	NA	$b_1 \prec a_4$
b_2	$a_1 \prec b_2$	$a_2 \prec b_2$	NA	NA
b_3	$a_1 \prec b_3$	$a_2 \prec b_3$	$a_3 \prec b_3$	$a_4 \prec b_3$

3.1.2 Probabilistic Skyline

The concept of the probabilistic skyline is first introduced in [84] for uncertain data. It indicates the probability of an object to be selected in the skyline and can be used to calculate workers' trustability in this work.

3.1.2.1 Dominance Relation Between Uncertain Objects

There are two objects A and B, where A contains u tuples (i.e., A = $\{a_1, a_2, ..., a_u\}$), and B contains v tuples(i.e., B = $\{b_1, b_2, ..., b_v\}$). For simplicity, we assume that each tuple in an object has the same probability to occur. For $a_i \in A$ and $b_j \in B$, the probability that A $\prec$ B can be calculated by Eq. (3.1) [84]:

$$Pr(A \prec B) = \frac{1}{u \times v} \sum_{i=1}^{u} |\{b_j \in B | a_i \prec b_j\}| \tag{3.1}$$

Example 1: In Fig. 3.1, A, B and C are three workers, and each of them has 4, 3, and 4 historical reviews respectively (as seen in Table 3.1). In this example, we consider that each review has two dimensions (e.g., accuracy of the sensing data, and the reasonability of the ask price), and the smaller values are preferable. From Table 3.1, we know that there are totally 7 times that A dominates B, they are $\{a_1 \prec b_1, a_1 \prec b_2, a_1 \prec b_3, a_2 \prec b_2, a_2 \prec b_3, a_3 \prec b_3, a_4 \prec b_3\}$. Based on Eq. (3.1), we can simply get $Pr(A \prec B) = \frac{7}{3\times4} = \frac{7}{12}$; B only dominates A with $\{b_1 \prec a_4\}$, so $Pr(B \prec A) = \frac{1}{3\times4} = \frac{1}{12}$.

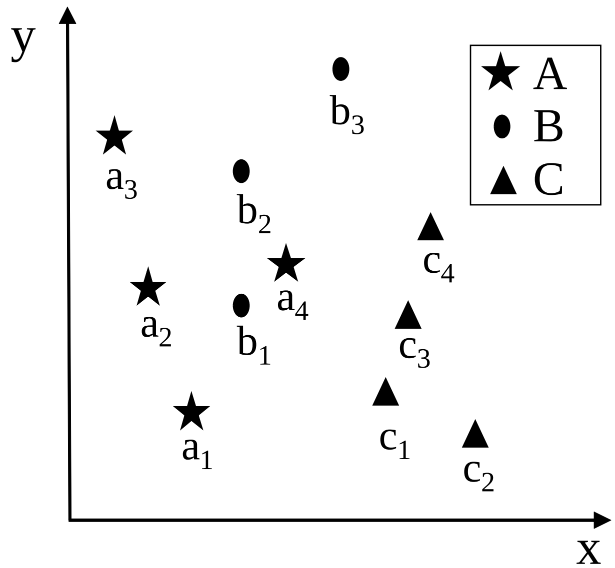


Figure 3.1: An example of three objects with different numbers of tuples for determining dominance relation and computing probabilistic skyline. The X and Y axis are two evaluation metrics for representing workers' sensing performance, for example, asking price and accuracy of the sensing data.

3.1.2.2 Probabilistic Skyline Computation

The probabilistic skyline value for object A can be calculated as:

$$PS\text{-}score = \frac{1}{|A|} \sum_{a_i \in A} Pr(a_i) \tag{3.2}$$

where for $a_i \in A$, $Pr(a_i)$ is the skyline probability for tuple a_i and can be calculated by Eq. (3.3).

$$Pr(a_i) = \prod_{A} \left(1 - \frac{|b_j \in B | b_j \prec a_i|}{|B|}\right) \tag{3.3}$$

Example 2: Let consider the example in Fig. 3.1 again. For worker B, b_1 is dominated by a_1, b_2 is dominated by a_1 and a_2, b_3 is dominated by a_1, a_2, a_3, a_4, and c_2. Then $Pr(b_1) = 1 - \frac{1}{4} = \frac{3}{4}$, $Pr(b_2) = 1 - \frac{2}{4} = \frac{1}{2}$, $Pr(b_3) = (1 - \frac{4}{4}) \times (1 - \frac{1}{4}) = 0$. The probabilistic skyline value for worker B can be calculated as $PS\text{-}score = \frac{1}{|B|}(Pr(b_1) +$

$Pr(b_2) + Pr(b_3)) = \frac{1}{3} \times (\frac{3}{4} + \frac{1}{2} + 0) = \frac{5}{12}$. Similarly, we can compute the probabilities that A and C are in the skyline.

3.1.3 Probabilistic Skyline over Sliding Windows

Probabilistic skyline can compute the likelihood that one object to be selected as the skyline points [84] [61]. However, in many real-time monitoring platforms, objects appear sequentially. The arrival time and active time-span of the objects are essential aspects for deciding the dominance relationship between different objects. [133] studied the problem of operating probabilistic skyline over sliding windows, which can be described as follows.

Let $\mathcal{W}_c = \{w_1, w_2, ..., w_{N_{win}}\}$ denotes the subset of current N_{win} workers. Each worker w_i is associated with a trustability score $T_i \in (0, 1)$ for $i \in [1, N_{win}]$. In addition, we define $\overline{T}_i = 1 - T_i$ as the complement of workers' trustability and $\overline{T}_i \in (0, 1)$ as well. According to [133], we use $\overline{\mathrm{Pr}}(w_i)$ to denote the probability that w_i is not dominated by any other workers in $\mathcal{W}_c$, so $\overline{\mathrm{Pr}}(w_i)$ can be calculated as:

$$\overline{\mathrm{Pr}}(w_i) = \prod_{\substack{w_k \in \mathcal{W}_c, \\ i \neq k, w_k \prec w_i}} (1 - T_k) \cdot 1^{N_i} = \prod_{\substack{w_k \in \mathcal{W}_c, \\ i \neq k, w_k \prec w_i}} \overline{T}_k \cdot 1^{N_i}. \tag{3.4}$$

Let $\mathcal{W}_{c/i}$ denote the set of workers in $\mathcal{W}_c$ (not include w_i) who do not dominate w_i, i.e., $\mathcal{W}_{c/i} = \{w_k | w_k \in \mathcal{W}_c, w_k \prec w_i, k \neq i\}$. In Eq. (3.4), N_i is the cardinality of $\mathcal{W}_{c/i}$, i.e., $N_i = |\mathcal{W}_{c/i}|$.

Let $\mathrm{Pr}^{\mathsf{sky}}(w_i)$ denote w_i's probabilistic skyline value that indicates the possibility that w_i appears in the skyline of $\mathcal{W}_c$. $\mathrm{Pr}^{\mathsf{sky}}(w_i)$ is calculated based on workers' trustability, which is shown in the following equation

$$\mathrm{Pr}^{\mathsf{sky}}(w_i) = T_i \cdot \overline{\mathrm{Pr}}(w_i) = T_i \cdot \prod_{\substack{w_k \in \mathcal{W}_c, \\ i \neq k, w_k \prec w_i}} \overline{T}_k \cdot 1^{N_i}. \tag{3.5}$$

To better present how probabilistic skyline over sliding windows can be used for worker selection, a motivating example is given as follows.

Example 3: A city government plans to recruit a number of workers to monitor local air quality for one month. The skyline computation depends on worker's working experience and the expiry time of their sensing data. In addition, each worker is associated with a trustability score which is derived from historical reporting records, and can be used to indicate the trustability of this worker. Due to the large-scale and fast arrival of the sensing data, the platform needs to select workers efficiently over sliding windows. For simplicity, we assume that the platform only focuses on the most 5 recent sensing data points. In Table 3.2, each worker w_i' personal information is listed for $i \in [1, 6]$, including worker $\texttt{ID}_i$, arrival time $t_i^{\texttt{arr}}$, the life span of the sensing data $t_i^{\texttt{ls}}$, the expiry time of the sensing data $t_i^{\texttt{exp}}$ (i.e., $t_i^{\texttt{exp}} = t_i^{\texttt{arr}} + t_i^{\texttt{ls}}$), working experience $\mathcal{E}_i$, trustability T_i and the complement of trustability $\overline{T}_i$. Specifically, at time 9:10 am, w_1 to w_5 are the qualified candidates, since they are the only and most 5 recent workers, and none of their sensing data expires. However, when w_6 arrives at 9:11 am, this platform needs to decide which 5 workers can be kept in the system among the 6 candidates. Here, probabilistic skyline is computed based on Eq. (3.5) and is used to rank the qualification of workers. For instance, if we want to compute $\Pr^{\texttt{sky}}(w_3)$, we first know that w_3 is dominated by w_2, w_4, w_5 and w_6 (see in Fig. 3.1), then $\overline{\Pr}(w_3) = (1 - T_2) \cdot (1 - T_4) \cdot (1 - T_5) \cdot (1 - T_6) = \overline{T}_2 \cdot \overline{T}_4 \cdot \overline{T}_5 \cdot \overline{T}_6 = 0.0149$, and finally $\Pr^{\texttt{sky}}(w_3) = T_3 \cdot \overline{\Pr}(w_3) = 0.9500 * 0.0149 = 0.0141$. After computing all workers' probabilistic skyline values, the worker with the minimum value will be removed from the system, while the rest of the workers will be kept for conducting $\mathcal{M}$.

From this example, we can see that probabilistic skyline computation over sliding windows is a practical and useful approach for worker selection. It can consider the trade-off between multiple criteria (e.g., working experience, recency of the sens-

Table 3.2: An example of air quality mointoring system

ID	$t^{\texttt{arr}}$	$t^{\texttt{ls}}$	$t^{\texttt{exp}}$	$\mathcal{E}$	T	$\overline{T}$
w_1	9:01 am	11 min	9:12 am	21	0.8500	0.1500
w_2	9:02 am	15 min	9:17 am	15	0.8500	0.1500
w_3	9:03 am	10 min	9:13 am	14	0.9500	0.0500
w_4	9:05 am	12 min	9:17 am	26	0.5500	0.4500
w_5	9:08 am	10 min	9:18 am	27	0.4500	0.5500
w_6	9:11 am	16 min	9:27 am	18	0.6000	0.4000

ing data, longer sojourn time) and discount the dominating criteria with too low trustability. Therefore, the workers with higher probabilistic skyline values are more reliable and trustable for fulfilling the task.

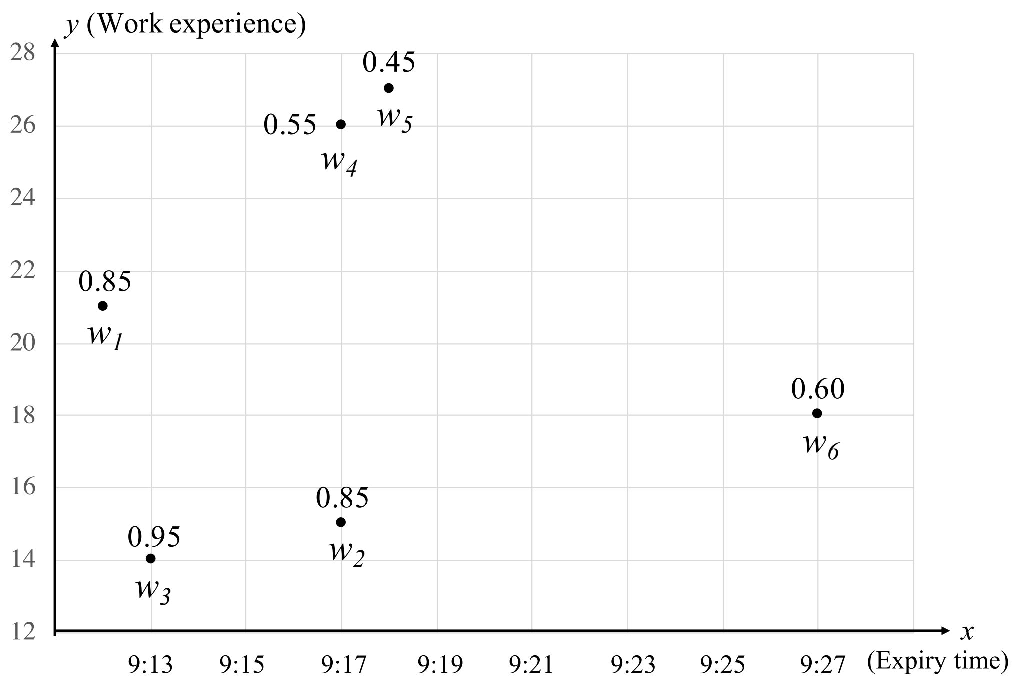


Figure 3.2: The illustration of the example in Table 3.2 of calculating probabilistic skyline over sliding window based on working experience and life span

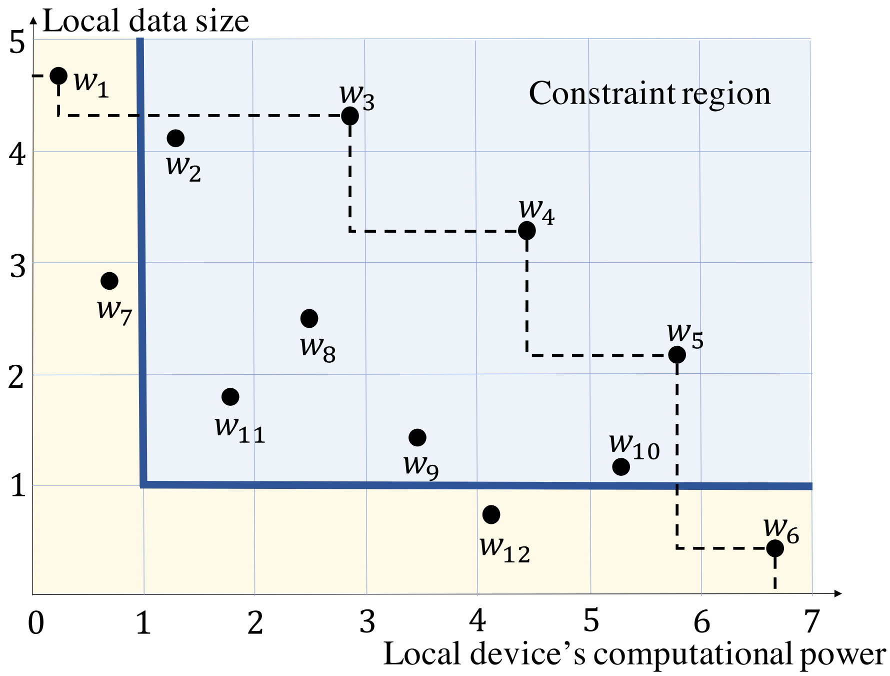


Figure 3.3: An simple example of skyline, G-skyline, C-skyline, and CG-skyline with 12 workers in terms of local data size and device's computational power

3.1.4 Group Skyline, Constrained Skyline, and Constrained Group Skyline

In this section, we use MCS workers as examples to illustrate Group skyline (G-skyline) [61, 60], and Constrained skyline (C-skyline) [14]. After that, we propose a novel skyline variant called Constrained Group skyline (CG-skyline), which can retrieve skyline groups with specific dimension restrictions on the data.

Definition 3.1.1. *(k-**point G-skyline**) Let $G = \{w_1, w_2, ..., w_k\}$, $G' = \{w'_1, w'_2, ..., w'_k\}$ are two different groups with size k where $k \leq K$ and $G, G' \subseteq \mathcal{W}$. Based on [61], we say G g-**dominates** G', denoted by $G \prec_g G'$, if we can find two permutations of k points for G and G', i.e., $G = \{w_{u1}, w_{u2}, ..., w_{uk}\}$ and $G' = \{w'_{v1}, w'_{v2}, ..., w'_{vk}\}$, such that $w_{ui} \preceq w'_{vi}$ for all $i \in [1, k]$, and there exists at least one i for $w_{ui} \prec w'_{vi}$. Consequently, k-point G-skyline consists of all the skyline groups with k workers that are not g-dominated by other groups with the same size.*

Example 3.1.1. *In Fig. 3.3, if we consider $k = 3$, then for two groups $G = \{w_3, w_4, w_5\}$ and $G' = \{w_2, w_9, w_{10}\}$, we have $G \prec_g G'$ due to $w_3 \prec w_2$, $w_4 \prec w_9$, and $w_5 \prec w_{10}$. Moreover, $G = \{w_3, w_4, w_5\}$ is 3-point G-skyline group since there exists no other group with the same size g-dominates G.*

Definition 3.1.2. *(**C-skyline**) Given a set of d-dimensional workers $\mathcal{W}$, let $\mathcal{C} = \{\mathtt{Con}_1, \mathtt{Con}_2, ..., \mathtt{Con}_d\}$ denote a set of d constraints, where each $\mathtt{Con}_i$ for $i \in [1, d]$ is either a range $[min_i, max_i]$ along a dimension or an empty set $\emptyset$ indicating no constraint in that dimension. Typically, the conjunction of all constrains in $\mathcal{C}$ forms a **constraint region** in the d-dimensional attribute space. C-skyline [83, 14, 21] retrieves all the skyline points in the constraint region.*

Definition 3.1.3. *(k**-point CG-skyline**) Given a set of d-dimensional workers $\mathcal{W}$, and a set of constraints $\mathcal{C} = \{\mathtt{Con}_1, \mathtt{Con}_2, ..., \mathtt{Con}_d\}$, k-point CG-skyline consists of all the skyline groups in the constraint region with k workers that are not g-dominated by other groups with the same size.*

Example 3.1.2. *In Fig. 3.3, $\mathcal{C} = \{\mathtt{Con}_1, \mathtt{Con}_2\}$ is a constraint set, where $\mathtt{Con}_1 = [1, 5]$ and $\mathtt{Con}_2 = [1, 7]$. The corresponding constraint region is the light blue area in the figure. The C-skyline workers are w_3, w_4, w_5 (since w_1 and w_6 are not in the constraint region). Besides, group $G = \{w_1, w_4, w_5\}$ is a 3-point G-skyline group but not a 3-point CG-skyline group because worker w_1 is not in the constraint region.*

3.2 Cryptography Technology

3.2.1 Bilinear Pairing

Let $\mathbb{G}, \mathbb{G}_T$ be two cyclic groups of the same prime order q. And g is a generator of group $\mathbb{G}$. Suppose $\mathbb{G}$ and $\mathbb{G}_T$ are equipped with a pairing, and a non-degenerated and efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$ has the following properties [147, 64]:

• Bilinearity. For all $x, y \in \mathbb{G}$, then $e(x^a, y^b) = e(x, y)^{ab}$ for $a, b \in \mathbb{Z}_q^*$.

• Non-degeneracy. $e(g, g) \neq 1_{\mathbb{G}_T}$.

• Computability. For all $x, y \in \mathbb{G}$, there exists an efficient algorithm to compute $e(x, y) \in \mathbb{G}_T$.

3.2.2 Paillier Cryptosystem

The Paillier Cryptosystem consists of key generation $\texttt{KeyGen}(\kappa)$, encryption $\texttt{E}(pk, m)$, and decryption $\texttt{D}(sk, c)$.

• $\texttt{KeyGen}(\kappa)$: Given a security parameter κ, two big primes p and q are first chosen such that $|p| = |q| = \kappa$. Then, $n = pq$ and $\lambda = lcm(p-1, q-1)$ are computed. Given a function $L(u) = \frac{u-1}{n}$, let g denote a generator of $\mathbb{Z}_{n^2}^*$, $\mu = (L(g^\lambda \mod n^2))^{-1}$. The public key is $pk = (n, g)$, and the corresponding private key is $sk = (\lambda, \mu)$.

• $\texttt{E}(pk, m)$: Given a plaintext $m \in \mathbb{Z}_n$ and a random number $r \in \mathbb{Z}_n^*$, the ciphertext can be calculated as $c = \texttt{E}(pk, m) = g^m \cdot r^n \mod n^2$.

• $\texttt{D}(sk, c)$: Given the ciphertext $c \in \mathbb{Z}_{n^2}^*$, m can be recovered as $m = \texttt{D}(sk, c) = L(c^\lambda \mod n^2) \cdot \mu \mod n$.

In Paillier cryptosystem, the product of two ciphertexts can be decrypted to the sum of their corresponding plaintexts, i.e.,

$$\texttt{D}(sk, \texttt{E}(pk, m_1) \cdot \texttt{E}(pk, m_2) \bmod n^2) = m_1 + m_2 \bmod n. \tag{3.6}$$

3.2.3 The Modified Paillier Public Key Encryption (PKE)

The modified Paillier PKE is proposed in [10], which can achieve proxy re-encryption property and homomorphic property. It consists of four algorithms, namely, key generation $\mathsf{Paillier.KeyGen}(\kappa)$, encryption $\mathsf{Paillier.Enc}(pk, m)$, decryption $\mathsf{Paillier.Dec}(sk, c)$, and proxy re-encryption Paillier.Pro.

- Paillier.KeyGen(κ): Given a security parameter $\kappa \in \mathbb{Z}^+$, two large primes p and q are randomly chosen, such that $|p| = |q| = \kappa$, and $n = pq$. Let $g = a^{2n}$ where a is randomly chosen $\in \mathbb{Z}^*_{n^2}$, and g is with the order of $(p-1)(q-1)/2$. Let s is a random number $\in [1, n^2/2]$. Paillier.KeyGen(κ) outputs the public key $pk = (n, g, h = g^s \bmod n^2)$ and secret key $sk = s$.

- Paillier.Enc(pk, m): Given pk from Paillier.KeyGen(κ) and a message $m \in \mathbb{Z}_n$, r is randomly chosen $\in [1, n/4]$. The message m can be encrypted as $c =$ Paillier.Enc$(pk, m) = (C_1, C_2)$, where $C_1 = g^r \bmod n^2$ and $C_2 = h^r(1+mn) \bmod n^2$.

- Paillier.Dec(sk, c): Given sk from Paillier.KeyGen(κ) and a ciphertext c, where $c =$ Paillier.Enc$(pk, m) = (C_1, C_2)$, the message m can be decrypted as $m =$ Paillier.Dec$(sk, c) = \mathcal{L}(C_2/C_1^s)$, where $\mathcal{L}(u) = \frac{(u-1) \bmod n^2}{n}$ for all $u \in \{u < n^2 | u = 1 \bmod n\}$.

- Paillier.Pro: Assume that the secret key s is split into two shares s_1 and s_2, such that $s = s_1 + s_2$. An encrypted message $c =$ Paillier.Enc$(pk, m) = (C_1, C_2)$ can be partially decrypted to a pair of ciphertexts $c^* = (C_1^*, C_2^*)$ using s_1, such that $C_1^* = C_1, C_2^* = (C_2/C_1^{s_1}) \bmod n^2$. The message m can be decrypted by $m =$ Paillier.Dec(s_2, c^*).

3.2.4 The Symmetric Homomorphic Encryption (SHE)

SHE supports efficient homomorphic addition and multiplication. It has been proved to be indistinguishable chosen plaintext attack (IND-CPA) secure [144]. Basically, SHE comprises three algorithms, namely key generation `KeyGen`(k_0, k_1, k_2), encryption `E`(sk), and decryption `D`(sk).

• $\texttt{KeyGen}(k_0, k_1, k_2)$: Given three security parameters $\{k_0, k_1, k_2\}$ satisfying $k_1 \ll k_2 < k_0$, the key generation algorithm first generates the secret key $sk = (p, q, \mathcal{L})$, where p and q are two large prime numbers with $|p| = |q| = k_0$ and $\mathcal{L}$ is a random number with $|\mathcal{L}| = k_2$. Then, the algorithm computes $\mathcal{N} = pq$. Finally, the algorithm outputs the public parameters $pp = (k_0, k_1, k_2, \mathcal{N})$ and the secret key $sk = (p, \mathcal{L})$. The basic message space $\mathcal{M}$ is $[-2^{k_1-1}, 2^{k_1-1})$.

• Encryption $\texttt{E}(m)$: Given the secret key sk, a message $m \in \mathcal{M}$ can be encrypted as $c = \texttt{E}(m) = (r\mathcal{L} + m)(1 + r'p) \mod \mathcal{N}$, where $r \in \{0,1\}^{k_2}$ and $r' \in \{0,1\}^{k_0}$ are two random numbers.

• Decryption $\texttt{D}(c)$: Given the security key sk, the original message m can be recovered by $m = \texttt{D}(c) = (c \mod p) \mod \mathcal{L}$.

According to [68], the SHE technique satisfies the following homomorphic addition and multiplication properties.

• *Homomorphic addition-I*: Two ciphertexts $\texttt{E}(m_1)$ and $\texttt{E}(m_2)$ satisfy that $\texttt{E}(m_1) + \texttt{E}(m_2) \rightarrow \texttt{E}(m_1 + m_2)$.

• *Homomorphic multiplication-I*: Two ciphertexts $\texttt{E}(m_1)$ and $\texttt{E}(m_2)$ satisfy that $\texttt{E}(m_1) \cdot \texttt{E}(m_2) \rightarrow \texttt{E}(m_1 \cdot m_2)$.

• *Homomorphic addition-II*: A ciphertext $\texttt{E}(m_1)$ and a plaintext m_2 satisfy that $\texttt{E}(m_1) + m_2 \rightarrow \texttt{E}(m_1 + m_2)$.

• *Homomorphic multiplication-II*: A ciphertexts $\texttt{E}(m_1)$ and a plaintext m_2 ($m_2 > 0$) satisfy that $\texttt{E}(m_1) \cdot m_2 \rightarrow \texttt{E}(m_1 \cdot m_2)$.

The correctness and proof of the above homomorphic properties are given in [68, 144].

3.2.5 Bloom Filter

A Bloom Filter (BF) is a data structure that can be used to check whether an element is definitely not or possibly in a set of elements (as seen in Fig. 3.4). An empty BF contains a N-bit array $A[N]$ where all the bits are initially set to

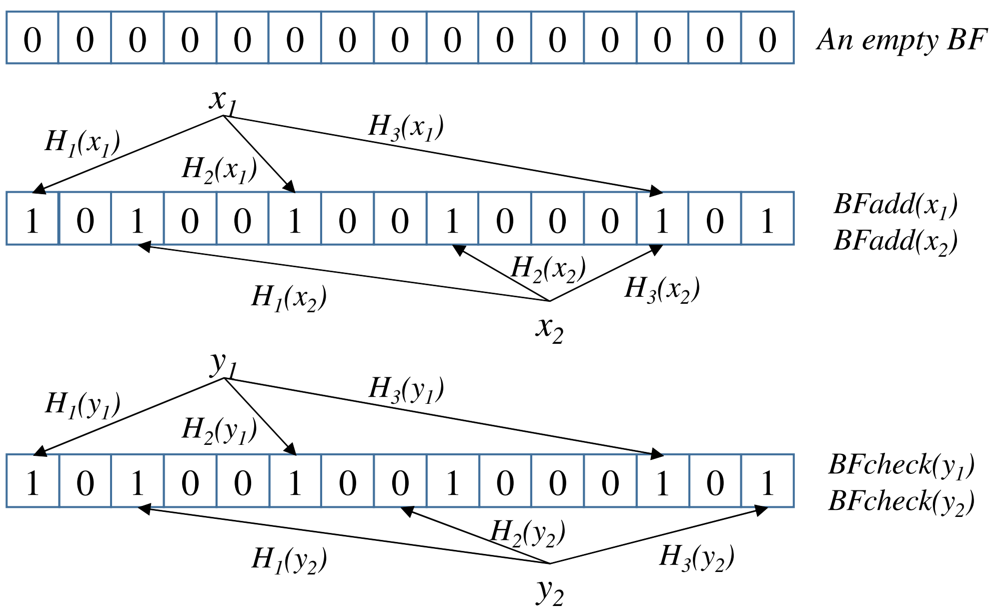


Figure 3.4: An example of three objects with different numbers of tuples for determining dominance relation and computing probabilistic skyline

0. There are two operations in BF, $BF.add(e)$, and $BF.check(e)$. In $BF.add(e)$, the BF randomly selects k independent hash functions $\{H_1, H_2, ..., H_k\}$, such that $H_i : \{0,1\}^* \rightarrow \{0, 1, 2, \cdots, N-1\}$ for $i \in [1, k]$. To add an element e, the BF assigns $A[H_i(e)] = 1$ for $i \in [1, k]$. In $BF.check(e)$, the BF can check whether an element e is included in the set or not by computing all the values of $A[H_i(e)]$. If all the values are set to 1, then we assume that e is in the set. Otherwise, e is definitely not included in the set. It worth noting that a false positive may happen even though all the values are 1. In order to build an optimal BF with lowest false positive rate, the number of hash functions k can be calculated as $k = \frac{N}{M} \ln 2$, where M is the number of elements that could be added.

3.3 Concluding Remarks

In this chapter, we have discussed the related background information about MCS applications and the essential technologies including different variants of skyline queries and some classic cryptography concepts and algorithms. These technologies will be used as the fundamental knowledge in this thesis. In the following chapters, we will present several scheme/protocols to address the major issues in MCS worker selection.

Chapter 4

Secure and Efficient Probabilistic Skyline Computation for Worker Selection in MCS

4.1 Introduction

Worker selection has been regarded as one of the fundamental issues in MCS, since the reliability of the workers and the trustability of their sensing data play significant roles in the service quality [135]. Recently, there has been extensive research on studying worker selection in MCS services [35, 45, 107, 111, 44, 34]. However, most of them evaluate workers based on the information provided by the workers. Such studies are infeasible for determining workers' real qualifications if there are strategic and selfish workers who try to maximize their profits by providing inaccurate information. In real-world MCS applications, workers' sensing performance is always task-by-task due to the limitations of sensing equipment and the dynamics of sensing conditions. Therefore, it is significant to measure workers' trustability and reliability based on their previous unstable behaviors. In addition, workers' personal information is inevitable to be shared; thus their privacy could be exposed during the process of worker selection. As a result, an ideal MCS platform should take workers' trustability into account, and preserve workers' privacy as well.

Motivated by the issues mentioned above, in this chapter, we propose an effective and privacy-preserving worker selection scheme for MCS services. Our proposed scheme is inspired by probabilistic skyline computation [84], which is useful and practical in multi-criteria decision analysis. Specifically, we design a privacy-preserving approach for securely calculating workers' probabilistic skyline value based on their historical reviews, and this value is considered as workers' trustability and will be used for worker selection. Specifically, the main contributions of this chapter are three-fold as follows.

- First, we propose a skyline based scheme for worker selection in MCS applications. To the best of our knowledge, this is the first skyline based approach for MCS studies. Based on the nature of skyline computation, the selected workers are not dominated by other workers on all the considered attributes. This property has not been achieved in the previous worker selection schemes in MCS applications.

- Second, we design a privacy-preserving probabilistic skyline technique for calculating workers' trustability based on their past performance. The method can aggregate the fluctuated historical reviews, and output a reliable trustability value for the workers. To protect workers' personal information from being disclosed during the calculation, we design a non-interactive encrypted integer comparison protocol, namely *NIEC*, which can compare the relation between two integers in a certain range without revealing their real values and even the difference of the values.

- Third, we analyze the security of the proposed scheme and show its privacy-preserving effectiveness for workers' personal information. Moreover, we conduct extensive performance evaluations in the experiment, which validates the efficiency and reliability of the process of worker selection.

The remainder of this chapter is organized as follows. In Section 4.2, we introduce our system model, security model and design goals. In Section 4.3, we propose our scheme in details. Then in Section 4.4 and Section 4.5, we present the security analysis and performance evaluation, followed by the related work in Section 4.6. Finally, we recap the conclusions in Section 4.7.

4.2 Models and Design Goals

In this section, we formalize our system model, security model, and identify our design goals. The proposed model is shown in Fig. 4.1. Table 4.1 shows a summary of notations used in this chapter.

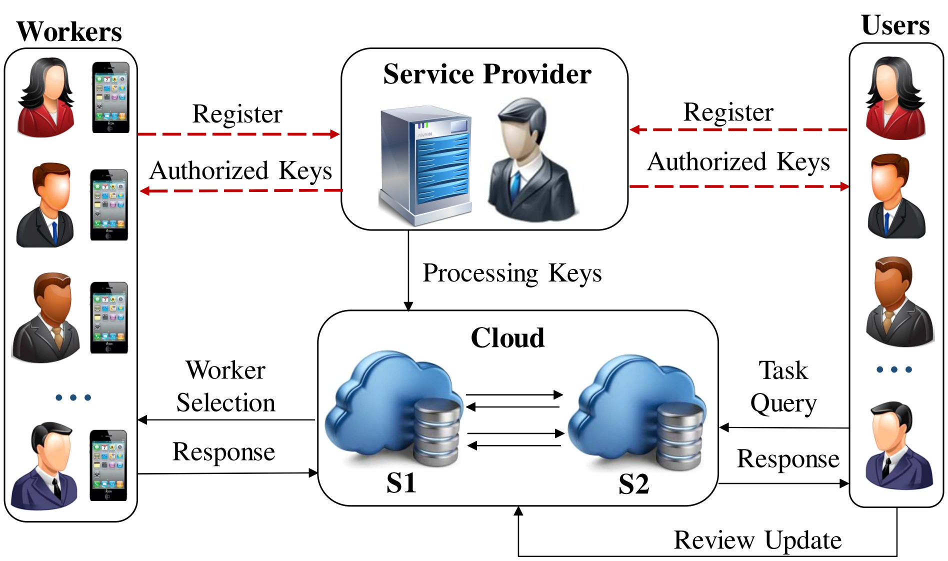


Figure 4.1: The system model under consideration in chapter 4.

4.2.1 System Model

Our system model mainly consists of four entities, namely a service provider (SP), a Cloud platform with two servers $\mathcal{CS} = \{S_1, S_2\}$, some users who request the task $\mathcal{U} = \{u_1, u_2, \cdots\}$, and some workers $\mathcal{W} = \{w_1, w_2, \cdots\}$.

• Service Provider (SP): SP is the service organizer who is responsible for bootstrapping the entire system. SP generates and distributes keys to different authorized entities, so that a certain task can be completed cooperatively. After that, SP just stays offline.

• Cloud Servers $\mathcal{CS} = \{S_1, S_2\}$: There are two Cloud servers (S_1, S_2) in our system model. After receiving a user's MCS task query Q, S_1 and S_2 will work together to select a subset of reliable workers based on workers' asking price ρ, task similarity μ, and probabilistic skyline value *PS-score*. After each sensing task is completed, S_1 and S_2 will work together to update each worker's *PS-score* based on users' reviews in an offline manner.

• Users $\mathcal{U} = \{u_1, u_2, \cdots\}$: In MCS services, authorized users are the task initiators. At first, a user sends a task query Q to the Cloud servers. After receiving the sensing data, the user is required to provide reviews for evaluating workers' performance. The metrics in the reviews may include the accuracy of the sensing data and the reasonability of the asking price.

• Workers $\mathcal{W} = \{w_1, w_2, \cdots\}$: In MCS services, workers are the participants who wish to conduct certain MCS task. They need to periodically send their approximate locations Δ_w to S_1. In order to be selected, workers are required to submit their sensing information to $\mathcal{CS}$, e.g., their asking price for the task ρ, and the task similarity μ.

4.2.2 Security Model

In our security model, we consider the SP is trustable. However, S_1 and S_2 are semi-honest, which means both of them strictly follow the protocol procedure, yet may be curious to learn additional information, i.e., workers' historical sensing performance. In addition, there is no collusion between S_1 and S_2. We assume that the users in our model are honest, and their reviews for assessing workers' performance are correct and unbiased. For the workers, we assume that they are strategic and selfish for maximizing their profits. However, in order to be selected by the platform again in the future, they still need to provide the correct information and the reasonable asking cost as much as possible. It is worth noting that there may exist outside attackers who can exploit the vulnerabilities of the MCS system, but they are beyond the scope of this thesis, and will be exploited in our future work.

4.2.3 Design Goals

In this work, our goal is to select reliable and suitable workers for conducting a sensing task. In specific, the following two objectives should be satisfied.

• *Privacy preservation:* Our work needs to consider the leakage of workers' privacy since a lot of sensitive information could be disclosed and inferred from the designed model. In the process of worker selection, it is unavoidable to share workers' personal information. Consequently, there may exist adversaries who are highly motivated to deduce workers' historical sensing performance [45], and a hostile MCS platform can exploit workers' sensing capacity for unfair competition. Therefore, each worker's historical reviews should not be disclosed to S_1, S_2, and other workers.

• *Efficiency:* An efficient worker selection process is significant for supporting real-world MCS applications, and we need to keep the computational cost of skyline query as low as possible. Given a set of workers with size δ, the time complexity for traditional skyline computation is $O(\delta^2)$ since each pair of workers should be

Table 4.1: The summary of important notations used in chapter 4

Notation	Definition
System Notations	
SP	Service provider
$CS = \{S_1, S_2\}$	Two Cloud servers
Worker and User Notations	
$\mathcal{U} = \{u_1, u_2, \cdots\}$	Authorized users
$\mathcal{W} = \{w_1, w_2, \cdots\}$	Registered workers
$\gamma = \{\gamma_1, \gamma_2, ..., \gamma_d\}$	A review with d dimensions
$[\gamma] = \{[\gamma_1], [\gamma_2], ..., [\gamma_d]\}$	An encrypted d-dimension review
PS-score	Worker's probabilistic skyline score
ρ	Worker's asking price
μ	Worker's task similarity
Encryption Notations	
m_1, m_2	Two positive integers
p, q	Two large prime numbers, $n = pq$
K	Authorized key
H_a	A cryptographic hash function
$pk = (g, h, n, H_a)$	Public key
$sk = (p)$	Processing key
$pub = \{pk, H_a\}$	System key
Simulation Notations	
l	Number of total workers
δ	Number of candidate workers
N_s	Number of selected skyline workers
N_a	Number of all skyline workers
t	The task similarity threshold
R	Location filtering threshold
v	Number of reviews per worker
d	Number of questions per review

compared. In this work, we aim to achieve a better efficiency such that the time complexity does only depend on N_s and the ratio $P_s = \frac{N_s}{\delta}$, where N_s is the number of skyline workers to be selected.

4.3 The Proposed Scheme

In this section, we present our skyline-based privacy-preserving scheme for worker selection in MCS services, which mainly consists of four phases: i) system initialization,

ii) task request and response, iii) worker selection, and iv) review update. Before delving the details of the proposed scheme, we first describe an encrypted integer comparison protocol called *NIEC* as follows, which can be used to non-interactively compare two integers without privacy leakage.

4.3.1 Non-Interactive Encrypted Integer Comparison Protocol (NIEC)

For two positive integers $m_1, m_2 \in [m_l, m_u]$ (e.g., the lower bound $m_l = 1$ and the upper bound $m_u = 100$), the goal of *NIEC* is to check whether $m_1 = m_2, m_1 > m_2$, or $m_1 < m_2$, without revealing their actual values. The overall settings of *NIEC* are shown in Fig. 4.2, which consists of *Data Owner*, *Encryptor*, *Preprocessor*, and *Comparator*. The details of *NIEC* are described as follows.

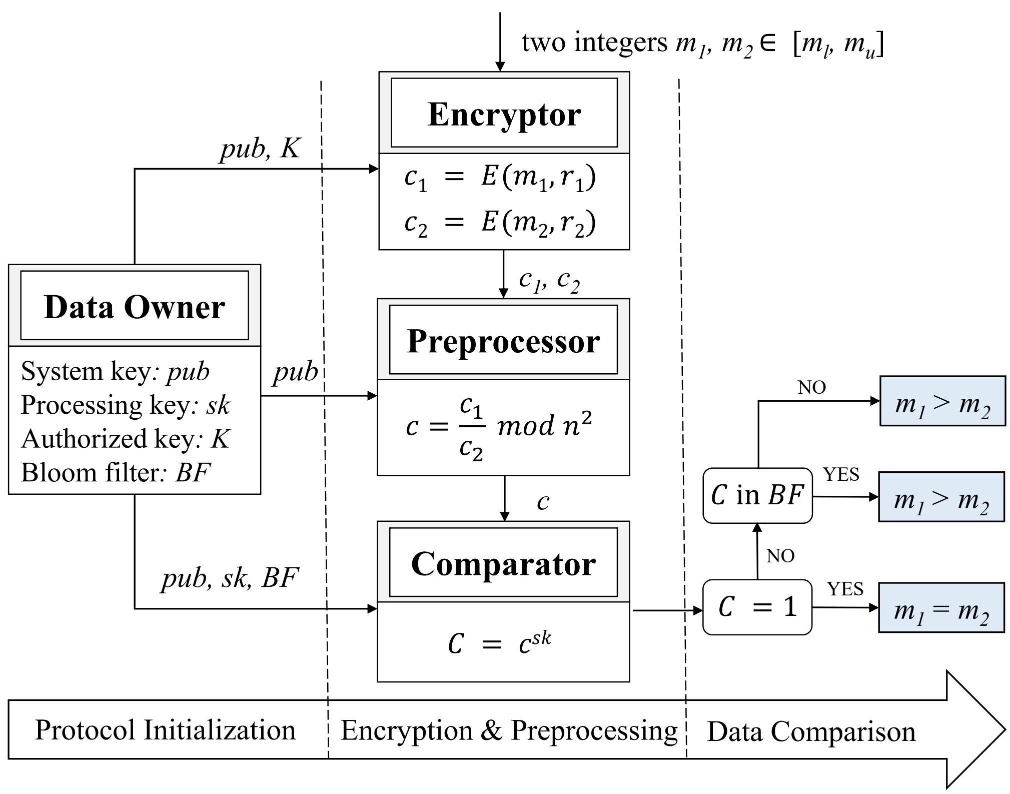


Figure 4.2: Overview of the encrypted integer comparison protocol NIEC

4.3.1.1 Protocol Initialization

Given a security parameter $\kappa \in \mathbb{Z}^+$, *Data Owner* randomly selects two large prime numbers p and q, such that the bit length $|p| = |q| = \kappa$ and let $n = pq$. Then *Data Owner* randomly chooses $g \in \mathbb{Z}^*_{n^2}$ to guarantee the order of g is n, i.e., $g^n \equiv 1 \; mod \; n^2$, and computes $h = g^q \; mod \; n^2$. The public key is $pk = (g, h, n)$, and the processing key is $sk = (p)$. After that, *Data Owner* creates an authorized key K and chooses a cryptographic hash function H_a, and uses Algorithm 1 to create a bloom filter BF. Finally, *Data Owner* publishes the system key $pub = \{pk, H_a\}$, and respectively distributes the authorized key K to *Encryptor*, the processing key sk and BF to *Data Comparator*.

Algorithm 1 BF generation for comparing two integers

Input: A N-bit length array $A[N]$ where all bits are initialized to 0, k independent hash functions $\mathcal{H} = \{H_1, H_2, \cdots, H_k\}$, $H_i : \{0,1\}^* \rightarrow \{0, 1, ..., N-1\}$ for each $H_i \in \mathcal{H}$, a cryptographic hash function H_a, public key $pk = (g, h, n)$, processing key sk, and authorized key K

Output: A bloom filter BF that can indicate $m_1 > m_2$ for two $m_1, m_2 \in [m_l, m_u]$

for $m_1 = m_l \; to \; m_u$ **do**
  for $m_2 = m_l \; to \; m_u$ **do**
    if $m_1 > m_2$ **then**
      $e = g^{p(H_a(m_1||K) - H_a(m_2||K))} \; mod \; n^2$
      for $j = 1 \; to \; k$ **do**
        set $A[H_j(e)] = 1$;
      end for
    end if
  end for
end for
return $A[N]$;

4.3.1.2 Data Encryption

For two integer $m_1, m_2 \in [m_l, m_u]$, the *Encryptor* first chooses two random numbers $r_1, r_2 \in \mathbb{Z}^*_n$ and then computes $c_1 = E(m_1, r_1) = g^{H_a(m_1||K)} h^{r_1} \bmod n^2$ and $c_2 = E(m_2, r_2) = g^{H_a(m_2||K)} h^{r_2} \bmod n^2$. Then it sends (c_1, c_2) to *Data Preprocessor*.

4.3.1.3 Data Preprocessing

Upon receiving (c_1, c_2) from *Encryptor*, *Data Preprocessor* computes $c = \frac{c_1}{c_2} \bmod n^2$, and then sends c to *Data Comparator*.

4.3.1.4 Data Comparison

Upon receiving c, *Data Comparator* computes $C = c^p \bmod n^2$. If $C = 1 \bmod n^2$, then $g^{p(H_a(m_1||K) - H_a(m_2||K))}\ mod\ n^2 = 1$, which means that $H_a(m_1||K) - H_a(m_2||K) = 0$ and thus $m_1 = m_2$. Else if $C \neq 1 \bmod n^2$, *Data Comparator* uses the bloom filter BF to check C, if C is in BF, then $m_1 > m_2$, otherwise, $m_1 < m_2$. Note that if there is no collusion between *Data Preprocessor* and *Data Comparator*, the messages m_1 and m_2 cannot be recovered from c_1 and c_2.

4.3.2 Description of Our Proposed Scheme

4.3.2.1 System Initialization

As SP is the service organizer, it is reasonable to consider SP is responsible for bootstrapping the entire system. The details of system initialization are as follows.

• First, SP plays the role of *Data Owner* in *NIEC* protocol, and generates public key $pk = (g, h, n)$, processing key $sk = (p)$, a bloom filter BF, an authorized key K, and chooses a hash function H_a by the approaches in Section 4.3.1. After that, SP will publish the system key $pub = \{pk, H_a\}$.

• Then, both users and workers need to register for the MCS platform through SP. After their registrations, SP distributes K to the registered users and workers, and sends (sk, BF) to the Cloud server S_2 in a secure way.

• SP maintains two global databases $\mathcal{W}$ and $\mathcal{U}$ for the registered workers and users respectively (see in Tables 4.2 and 4). After registering a new worker, SP assigns an ID to the worker, and adds him/her into $\mathcal{W}$. In addition, each worker's *PS-score* is

Table 4.2: The global worker table $\mathcal{W}$ in system initialization

W_{id}	$[\gamma]$	*PS-score*
w_1		0.5
w_2		0.5
w_3		0.5
...		...

Table 4.3: The global user table $\mathcal{U}$ in system initialization

U_{id}
u_1
u_2
u_3
...

initialized as 0.5 and is used as the trustability of this worker. $[\gamma]$ denotes workers' encrypted reviews and is initialized as empty. After registering all the workers and users, SP distributes $\mathcal{U}$ and $\mathcal{W}$ to Cloud server S_1.

• For being selected, each registered worker is required to regularly send his/her approximate location Δ_w to S_1. Using the geo-indistinguishability technique introduced in [2], workers' approximate location Δ_w can be computed by adding some random differential-privacy noise to their accurate positions. As a result, Δ_w can achieve differential-privacy property while still releasing approximate information.

After initializing the entire system, SP just stays offline, and will not participate in the MCS services.

4.3.2.2 Task Request and Response

This phase mainly contains three steps: *Users' task request*, *Location matching*, *Workers' task response*.

• *Users' task request*: At the beginning of an MCS task, a user sends a task query Q to S_1, which contains the task location Δ_q, and the task content Ω_q, i.e., $Q = \{\Delta_q, \Omega_q\}$.

• *Location matching*: After receiving the task query Q, S_1 finds the workers whose location information Δ_w matches the query's location requirement Δ_q. The matched workers are denoted as $\mathcal{W}_\Delta$, and $\mathcal{W}_\Delta \subseteq \mathcal{W}$.

• *Workers' task response*: In this step, S_1 announces the query Q to $\mathcal{W}_\Delta$. Each active worker in $\mathcal{W}_\Delta$ who wishes to conduct the task is required to send the feedback information Λ to S_1. $\Lambda = (\rho, \mu)$, where $\rho \in (0, 1)$ is worker's task similarity and $\mu \in (0, 1)$ is the worker's asking price for the task. ρ is used to indicate the relevancy between the task content and the workers' activities. Usually, a task content Ω_q is announced in the form of a sequence of keywords. A worker can then decide his/her task similarity by information retrieval models, such as text relevance analysis [19]. We assume that smaller ρ indicates higher similarity, so for both ρ and μ, the smaller values are more preferred. Then given a predefined threshold t, S_1 only selects the workers whose ρ is smaller than t.

After the similarity matching, S_1 joins the workers' *PS-score* with their Λ based on worker ID. The group of active workers is called $\mathcal{W}_\delta$ (see in Table 4.4), such that $\mathcal{W}_\delta \subseteq \mathcal{W}_\Delta$.

4.3.2.3 Worker Selection

In this step, we propose a new skyline computation algorithm (see in Algorithm 2), which can be employed to find N_s skyline workers efficiently. The general idea of our algorithm is introduced as follows. All workers in $\mathcal{W}_\delta$ have three attributes, namely asking price ρ, task similarity μ, and *PS-score*. First we pre-process and normalize *PS-score* with the equation: $\textit{PS-score} = \frac{\textit{PS-score}_{min}}{\textit{PS-score}}$, where $\textit{PS-score}_{min}$ is the minimum *PS-score* among all the workers. After that, all the three attributes are ranged from 0 and 1 and the smaller values are preferred. And we sort all the workers in a descending order by $S(w_i)$ where $S(w_i) = \rho_i + \mu_i + \textit{PS-score}_i$. Then the first worker (i.e., with minimum $S(w_i)$) is added in the skyline pool $S_{skyline}$ and is deleted from $\mathcal{W}_\delta$. Next, the worker with minimum $S(w_i)$ is selected and compared with the workers in $S_{skyline}$. If he/she is not dominated by all the workers in $S_{skyline}$, then add him/her into $S_{skyline}$ and delete him/her from $\mathcal{W}_\delta$. Otherwise, we can directly delete him/her from $\mathcal{W}_\delta$. This algorithm repeats the aforementioned steps for the remaining workers until either $\mathcal{W}_\delta$ is empty or there are N_s workers in $S_{skyline}$.

Table 4.4: The subset worker table $\mathcal{W}_\delta$ after workers' response

W_{id}	μ	ρ	*PS-score*
w_1	0.3	0.5	0.5
w_2	0.6	0.8	0.5
w_3	0.8	0.7	0.5
...	...	...	...

Table 4.5: The global worker table $\mathcal{W}$ after updating reviews

W_{id}	$[\gamma]$	*PS-score*
w_1	$[\gamma]_{1,1}$ ...	0.5
w_2	$[\gamma]_{2,1}$ ...	0.5
w_3	$[\gamma]_{3,1}$ ...	0.5
...	...	...

Consequently, S_1 announces all the skyline workers to conduct the task. After the sensing task is completed, S_1 collects workers' data and reports it to the user.

Algorithm 2 Skyline-based worker selection

Input: A worker database $\mathcal{W}_\delta$ such that $|\mathcal{W}_\delta| = \delta$, and each worker w_i in $\mathcal{W}_\delta$ can be represented as $w_i = (\rho_i, \mu_i, \textit{PS-score}_i)$;
A number N_s which is defined by users
Output: The skyline workers of $\mathcal{W}_\delta$
for $i = 1$ *to* δ **do**
  $\textit{PS-score} = \frac{\textit{PS-score}_{min}}{\textit{PS-score}}$;
  $S(w_i) = \rho_i + \mu_i + \textit{PS-score}_i$;
  $PQ.Insert((S(w_i), w_i))$;
end for
Set $S_{skyline}$ as an initially empty skyline set;
$count = 0$;
while PQ is not empty && $count < N_s$ **do**
  $w_{min} \leftarrow PQ.RemoveMin()$;
  if there is no worker s in $S_{skyline}$ such that $s \prec w_{min}$ **then**
    $S_{skyline}.Add(w_{min})$;
    $count = count + 1$;
  end if
end while
return $S_{skyline}$;

4.3.2.4 Review Update

This phase mainly contains two steps, *Review submission* and *Probabilistic skyline computation*.

• *Review submission*: After receiving the worker's data, the user is required to rate each worker's performance by filling in an evaluation form. Fig. 4.3 shows a simple

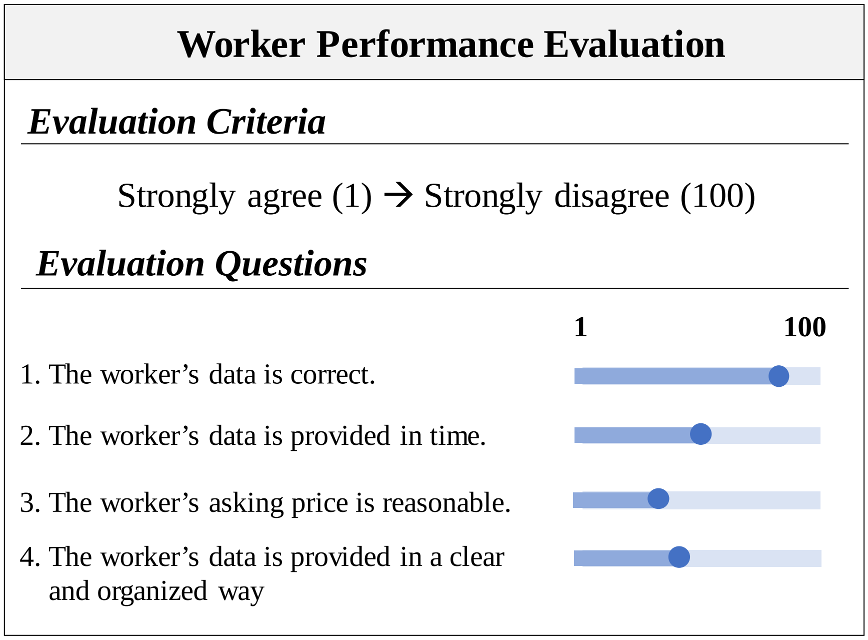


Figure 4.3: A simple example of the worker evaluation form with four evaluation questions. And for each question, a bootstrap slider can be used to answer users' satisfaction level (from 1 to 100) for a task in terms of a certain aspect.

example of a such form, which contains four evaluation questions. Users can use a bootstrap slider to select the satisfactory level of each question. The level is an integer $\in [m_l, m_u]$ (e.g., $m_l = 1, m_u = 100$), where m_l means strongly agree, and m_u means strongly disagree. So any review can be considered as a fixed-dimension vector $\gamma = \{\gamma_1, \gamma_2, ..., \gamma_d\}$, such that d is the number of questions, and $\gamma_i \in [m_l, m_u]$ is the integer that indicates the satisfaction degree.

Then, users need to encrypt each answer in γ using *pk* based on *NIEC* protocol. After encryption, each encrypted review $[\gamma]$ will be sent to S_1, and will be added in $\mathcal{W}$ by S_1. $[\gamma]$ can be used for computing and updating the *PS-score* for each worker. Table 6 shows the global worker table $\mathcal{W}$ after updating each worker's review. $[\gamma]_{i,j} = \{[\gamma_1]_{i,j}, [\gamma_2]_{i,j}, ..., [\gamma_d]_{i,j}\}$, where i means the worker ID, j means the $j^{\,th}$ review for the worker. Different workers may have different numbers of reviews.

• *Updating probabilistic skyline*: In our proposed scheme, the *PS-score* is updated in an offline manner. After receiving all the encrypted reviews from users, S_1 and S_2

Algorithm 3 RateDominance(γ_1, γ_2)

Note: given two original reviews $\gamma_1 = \{x_1, x_2, ..., x_d\}$ and $\gamma_2 = \{y_1, y_2, ..., y_d\}$, the corresponding encrypted reviews $[\gamma_1] = \{[x_1], [x_2], ..., [x_d]\}$ and $[\gamma_2] = \{[y_1], [y_2], ..., [y_d]\}$ are computed with public key pk in NIEC protocol in Section 4.3.1;
Input: the processing key $sk = p$; the bloom filter BF; an encrypted vector $V_c = (c_1, c_2, \cdots, c_d)$, where each $c_i = [x_i]/[y_i] \bmod n^2$, and $[x_i]$ belongs to $[\gamma_1] = ([x_1], [x_2], ..., [x_d])$, $[y_i]$ belongs to $[\gamma_2] = ([y_1], [y_2], ..., [y_d])$
Output: $0, 1, 2$
$token_{xy} = true$;
(indicating $\gamma_1 \prec \gamma_2$)
$token_{yx} = true$;
(indicating $\gamma_2 \prec \gamma_1$)
$count = 0$;
for $i = 1$ to d **do**
    $C_i = c_i^p \bmod n^2$;
    if C_i is in BF
    (indicating $x_i > y_i$, so $\gamma_1 \not\prec \gamma_2$) **then**
        $token_{xy} = false$;
    end if
    if $C_i == 1 \bmod n^2$
    (indicating $x_i = y_i$) **then**
        $count{+}{+}$;
    end if
    if C_i is not in BF && $C_i \neq 1 \bmod n^2$
    (indicating $x_i < y_i$, so $\gamma_2 \not\prec \gamma_1$) **then**
        $token_{yx} = false$;
    end if
end for
if $token_{xy} == true$ && $count < d$ **then**
    return 1 indicating $\gamma_1 \prec \gamma_2$
end if
if $token_{yx} == true$ && $count < d$ **then**
    return 2 indicating $\gamma_2 \prec \gamma_1$
end if
return 0 indicating $\gamma_1 \not\prec \gamma_2$ and $\gamma_2 \not\prec \gamma_1$

work together to update each worker's *PS-score* in offline. More specifically, given $[\gamma]_{w_i} = \{[\gamma_1]_{w_i}, [\gamma_2]_{w_i}, ..., [\gamma_d]_{w_i}\}$ from worker w_i and $[\gamma]_{w_j} = \{[\gamma_1]_{w_j}, [\gamma_2]_{w_j}, ..., [\gamma_d]_{w_j}\}$ from worker w_j, S_1 will first compute $V_c = (c_1, c_2, ..., c_d)$ where $c_s = \frac{[\gamma_s]_{w_i}}{[\gamma_s]_{w_j}}$ for $s \in (1, d)$. After computing for all pairs of workers from $\mathcal{W}$, S_1 sends V_c to S_2. Then S_2 can determine the dominance relation between all pairs of workers using Algorithm 3, update each worker's *PS-score* and fill in the 2-dimensional array $\mathcal{W}_s[l][l+1]$ in Algorithm 4. Finally, S_2 returns $\mathcal{W}_s[l][l+1]$ to S_1.

Algorithm 4 WorkerDominance(w_1, w_2)

Input: Two worker such that $w_1 = \{[\gamma]_{1,1}, [\gamma]_{1,2}, ..., [\gamma]_{1,u}\}$, $w_2 = \{[\gamma]_{2,1}, [\gamma]_{2,2}, ..., [\gamma]_{2,v}\}$, and $|w_1| = u, |w_2| = v$
Output: $Pr(w_1, w_2)$ indicating the probability that $w_1 \prec w_2$, and $Pr(w_2, w_1)$ indicating the probability that $w_2 \prec w_1$ $count_1, count_2 = 0$;
for $i = 1$ *to* u **do**
  for $j = 1$ *to* v **do**
    if RateDominance$([\gamma]_{1,i}, [\gamma]_{2,j}) = 1$
    (indicating $[\gamma]_{1,i} \prec [\gamma]_{2,j}$) **then**
      $count_1 + +$;
    end if
    if RateDominance$([\gamma]_{1,i}, [\gamma]_{2,j}) = 2$
    (indicating $[\gamma]_{2,j} \prec [\gamma]_{1,i}$) **then**
      $count_2 + +$;
    end if
  end for
end for
$Pr(w_1, w_2) = \frac{count_1}{u \times v}$;
$Pr(w_2, w_1) = \frac{count_2}{u \times v}$;
return $Pr(w_1, w_2)$ and $Pr(w_2, w_1)$;

4.4 Security Analysis

This section evaluates the security properties and shows how workers' reviews can be protected from being disclosed in the proposed scheme.

• *Privacy-preservation of users' reviews:* Our goal is to enable S_1 and S_2 to compute workers' *PS-score* without disclosing the exact answers of their reviews. On one hand,

Algorithm 5 *PS-score* Calculation

Input: The global worker table $\mathcal{W}$, $|\mathcal{W}| = l$
Output: A two-dimensional array $\mathcal{W}_s[l][l+1]$ that contains the updated *PS-score* for each worker in $\mathcal{W}$
Initialize $\mathcal{W}_s[l][l+1]$ to an initially empty 2-dimensional array;
for $i = 1$ *to* l **do**
    Initialize *PS-score* $= 1$;
    for $j = 1$ *to* l && $j \neq i$ **do**
        if $\mathcal{W}_s[i][j]$ is empty **then**
            $p_1, p_2 = \text{WorkerDominance}(w_i, w_j)$;
        end if
        PS-score = *PS-score* $\times$ $(1 - \mathcal{W}_s[i][j])$;
    end for
    $\mathcal{W}_s[i][l+1] =$ *PS-score* ;
end for
return $\mathcal{W}_s[l][l+1]$

according to the *NIEC* protocol, S_1 plays the role of *Data Preprocessor*, and has access to $pk = (g, h, n)$. For one orignial review answers m_1 and m_2, S_1 knows their corresponding ciphertexts $c_1 = g^{H_a(m_1||K)} h^{r_1} \bmod n^2$ and $c_2 = g^{H_a(m_2||K)} h^{r_2} \bmod n^2$. However, without knowing $sk = (p)$ and the random numbers (r_1, r_2), S_1 cannot remove the random factors $h^{r_1} \bmod n^2$ and $h^{r_2} \bmod n^2$ in the ciphertexts. Furthermore, S_1 has no information about H_a and K. As a result, S_1 cannot recover any plaintexts of the original reviews answers in the proposed scheme.

On the other hand, S_2 plays the role of *Data Comparator* in the *NIEC* protocol, and has access to $sk = (p)$. Once receiving $c = \frac{c_1}{c_2}$ from S_1, S_2 can calculate $C = c^p = g^{p(H_a(m_1||K) - H_a(m_2||K))}$ mod n^2. However, S_2 cannot obtain the original values of m_1 and m_2 for the following two reasons. (i) Only knowing $pk = (g, h, n)$ and $sk = (p)$, due to discrete logarithm problem, it is hard for S_2 to compute $p(H_a(m_1||K) - H_a(m_2||K))$. (ii) We know each review is in a certain range, e.g., $m_1, m_2 \in [m_l, m_u]$. Even though S_2 might know the lower boundary m_l and the upper boundary m_u of the reviews answers m_1 and m_2, without knowing H_a and K, S_2 cannot calculate $H_a(m_1||K)$ and $H_a(m_2||K)$. So S_2 cannot brute force the plaintexts of m_1 and m_2 by checking all their possibles values either. Therefore, the reviews are privacy-preserving for S_2 as well.

• *Privacy-preservation of the difference between users' reviews:* Given any two plaintexts m_1 and m_2, neither S_1 nor S_2 knows the difference of them, i.e., $m_1 - m_2$. On one hand, S_1 can only know $c = \frac{c_1}{c_2}$ mod $n^2 = g^{(H_a(m_1||K) - H_a(m_2||K))} h^{r_1 - r_2} \bmod n^2$. Without knowing H_a, K, h and random numbers (r_1, r_2), S_1 cannot determine $m_1 - m_2$ even thougn S_1 might know (m_l, m_u). In addition, given another pair of review answers m'_1 and m'_2, we assume that S_1 knows (m'_1, m'_2) and the corresponding ciphertexts $(c'_1, c'_2, c', c_1, c_2)$ such that $c'_1 = g^{H_a(m'_1||K)} h^{r'_1} \bmod n^2, c'_2 = g^{H_a(m'_2||K)} h^{r'_2} \bmod n^2$ and $c' = \frac{c'_1}{c'_2} = g^{(H_a(m'_1||K) - H_a(m'_2||K))} h^{r'_1 - r'_2} \bmod n^2$. However, S_1 cannot obtain whether $m'_1 - m'_2 = m_1 - m_2$ by simply comparing c and c' due to the exist of random factors $(h^{r_1} \bmod n^2, h^{r_2} \bmod n^2, h^{r'_1} \bmod n^2, h^{r'_1} \bmod n^2)$.

On the other hand, as we mentioned before, S_2 cannot recover $p(H_a(m_1||K) - H_a(m_2||K))$ from $C = g^{p(H_a(m_1||K) - H_a(m_2||K))}$ mod n^2, which means S_2 cannot determine $m_1 - m_2$ either. Next, let consider the following scenario. We assume for two pairs of plaintexts (m_1, m_2) and (m'_1, m'_2), S_2 knows (m'_1, m'_2, C, C') such that $C' = g^{p(H_a(m'_1||K) - H_a(m'_2||K))}$ mod n^2. Still, S_2 cannot decide whether $m'_1 - m'_2 = m_1 - m_2$ or not, because even though $C = C'$ might hold (i.e., $H_a(m'_1||K) - H_a(m'_2||K) = H_a(m_1||K) - H_a(m_2||K)$ holds), the differences of their original values are not the same, i.e., $m'_1 - m'_2 \neq m_1 - m_2$. Therefore, S_2 cannot obtain whether $m_1 - m_2 = m'_1 - m'_2$ by only checking whether $C = C'$ or not.

Furthermore, based on C and C', S_2 can calculate $\frac{C}{C'} = \frac{g^{p(H_a(m_1||K) - H_a(m_2||K))}}{g^{p(H_a(m'_1||K) - H_a(m'_2||K))}} \bmod n^2 = g^{p(H_a(m_1||K) - H_a(m_2||K) - H_a(m'_1||K) + H_a(m'_2||K))} \bmod n^2$. Even though $H_a(m_1||K) - H_a(m_2||K) - H_a(m'_1||K) + H_a(m'_2||K)$ might happen to be 0 and $\frac{C}{C'}$ might happen to be 1, S_2 cannot determine whether $m'_1 - m'_2 = m_1 - m_2$. This is because after using the hash function H_a, $m'_1 - m'_2 = m_1 - m_2$ does not mean $H_a(m_1||K) - H_a(m_2||K) = H_a(m'_1||K) - H_a(m'_2||K)$, and vice versa.

In summary, both the real values of the review answers and the difference of any pair of review answers are privacy-preserving for both S_1 and S_2 as long as there is no collusion between them. In addition, by introducing the hash function H_a and the authorized key K, our proposed scheme can effectively resist known plaintext attacks.

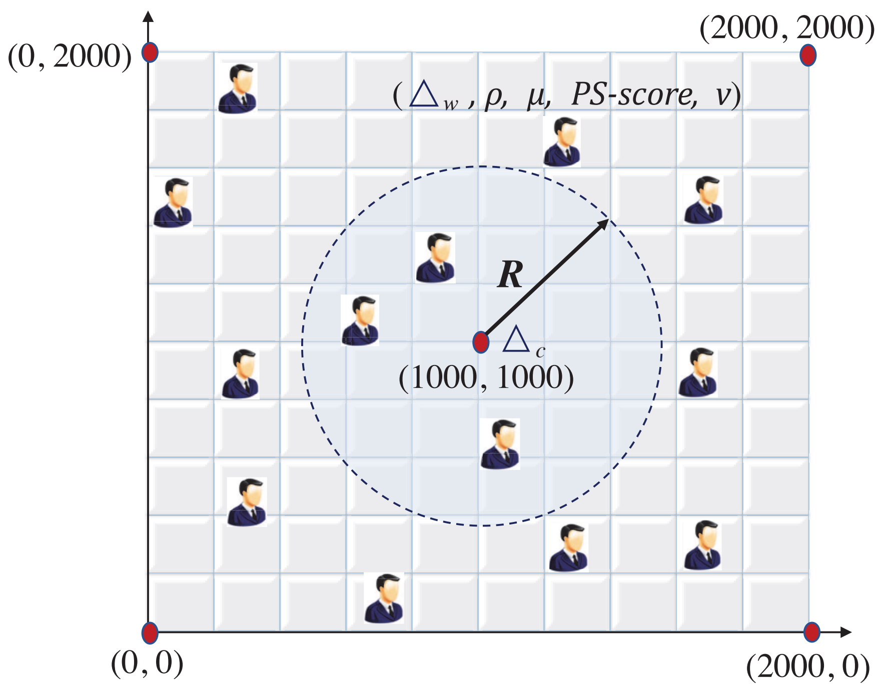
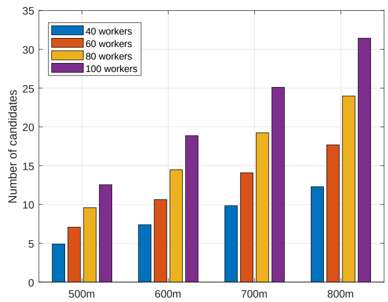


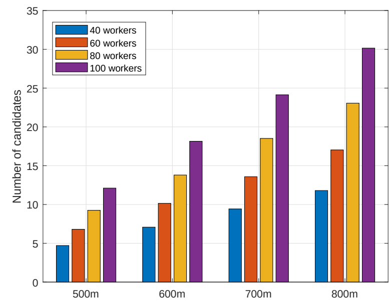
Figure 4.4: The overall illustration of the experimental simulation and parameter settings

4.5 Performance Evaluation

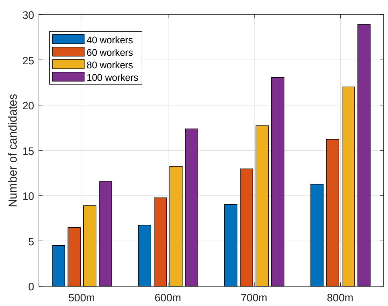
In this section, we study the effectiveness of our proposed scheme using a custom simulator built with Java programming language. First, we assess the performance of the proposed scheme theoretically in terms of storage and computational overhead. Second, a series of experiments are being conducted to investigate how our scheme's performance varies across different experimental settings. The performance metrics are i) the number of candidate workers, ii) the running time for selecting skyline



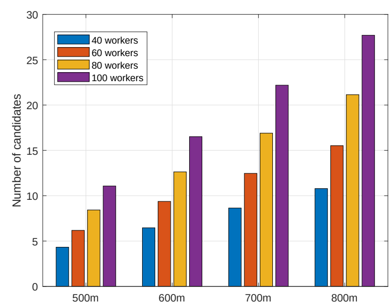
(a) The task similarity threshold t is set to 0.46, the total number of workers l varies from 40 to 100, the location filtering threshold R varies from 500m to 800m



(b) The task similarity threshold t is set to 0.44, the total number of workers l varies from 40 to 100, the location filtering threshold R varies from 500m to 800m



(c) The task similarity threshold t is set to 0.42, the total number of workers l varies from 40 to 100, the location filtering threshold R varies from 500m to 800m



(d) The task similarity threshold t is set to 0.40, the total number of workers l varies from 40 to 100, the location filtering threshold R varies from 500m to 800m

Figure 4.5: The number of candidate workers under with different experiment settings

workers and updating the *PS-score* for all the workers, and iii) the *PS-score* for the selected skyline workers. The running time can be used to examine the efficiency of the proposed scheme, the number of candidate workers and the *PS-score* for the selected workers are good indicators for the effectiveness of the scheme, i.e., how many workers can be selected from, and how reliable and trustable the selected workers are.

4.5.1 Theoretical Analysis

4.5.1.1 Storage Overhead

Assume that there are totally l registered workers in $\mathcal{W}$. Each worker w_i contains v reviews, and each review $\gamma = \{\gamma[1], \gamma[2], ..., \gamma[d]\}$ has d dimensions. Each dimension $\gamma[i]$ for $i \in (1, d)$ needs to be encrypted before outsourcing to $\mathcal{CS}$. So the overall storage overhead for all the encrypted reviews is $\sum_{i=1}^{l} \sum_{j=1}^{v} (d \cdot Len)$, where $Len = 2048$ is the length of ciphertext for each dimension.

4.5.1.2 Computational Cost

Next, we discuss the computation cost for the online skyline computation and the offline probabilistic skyline computation in our proposed scheme.

• *The computational cost for skyline computation*: The time complexity for building minHeap queue in Algorithm 2 for all the candidate workers is $O(\delta)$, where δ is the size of W_δ. Then, the time complexity for skyline computation depends on N_s and P_s, where $P_s = \frac{N_a}{\delta}$ for N_a is the number of all the skyline workers in W_δ. (i) When $N_s = 1$, we can just return the first worker in the queue, and the cost is $O(1)$. (ii) When $N_s = 2$, the expected computational cost is $2 \cdot P_s + 3 \cdot (1 - P_s) \cdot P_s + ... + l \cdot (1 - P_s)^{\delta - 2} \cdot P_s = \sum_{i=0}^{\delta-2} (i + 2) \cdot P_s \cdot (1 - P_s)^i$, which is equal to $1 + \frac{1}{P_s}$ when δ is quite large. Obviously, given a fixed δ, the cost is only inversely proportional to P_s. So our proposed algorithm is more efficient when N_s is small and P_s is large, and this conclusion can also be applied to the scenarios where $N_s > 2$.

• *The computational cost for probabilistic skyline computation*: To update all the workers' *PS-score*, the dominance relationship between every two workers needs to be determined. Assume every worker has v reviews and each review has d dimensions, in total *NIEC* needs to be run $\frac{l\times(l-1)}{2} \times v^2 \times d$ times. So the overall cost for probabilistic skyline updating depends on both l, v and d.

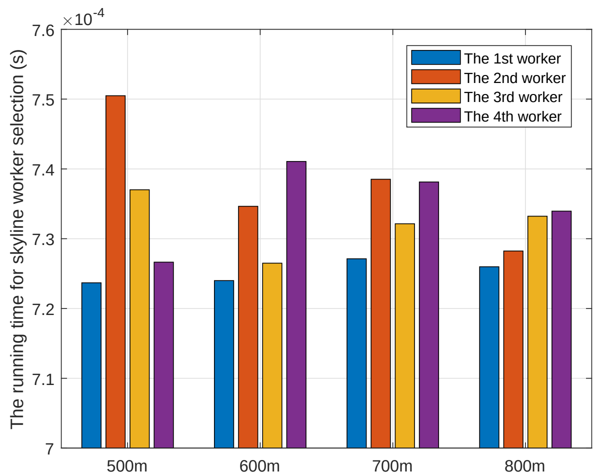
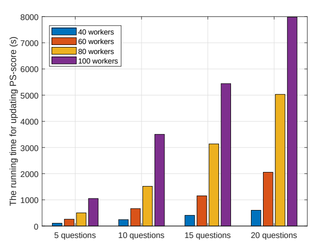


Figure 4.6: The running time for selecting skyline workers where the total number of workers l is 100, the number of selected skyline workers N_s is 4, the task similarity threshold t is 0.46, and location filtering threshold R varies from 500m to 800m

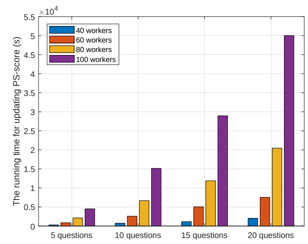
4.5.2 Simulation-based Analysis

4.5.2.1 Experiment Setup

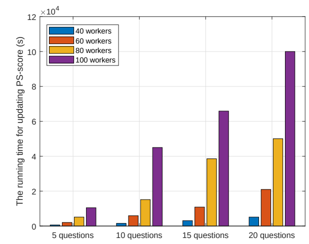
We simulate a sensing area $\mathcal{A}$ with the size of 2000m × 2000m and a collection of l workers with varied key parameters, which are demonstrated in Fig. 4.4. For each worker, the simulating parameters include his/her approximate location $\Delta_w = (x_w, y_w)$, task similarity ρ, asking price μ, *PS-score*, and the number of reviews v. We assume that all the MCS tasks are occurred at Δ_c, the centre location of $\mathcal{A}$ with the coordinate of (x_c = 1000m, y_c = 1000m). After a user requests a sensing



(a) The number of reviews v is set to 4, the total number of workers l varies from 40 to 100, the number of questions in each review d varies from 5 to 20



(b) The number of reviews v is set to 6, the total number of workers l varies from 40 to 100, the number of questions in each review d varies from 5 to 20



(c) The number of reviews v is set to 8, the total number of workers l varies from 40 to 100, the number of questions in each review d varies from 5 to 20

Figure 4.7: The running time for updating all workers' *PS-score* varies with different experiment settings

task, S_1 selects the matching workers based on a location filtering threshold R and task similarity filtering threshold t. More specifically, S_1 only selects a subset of workers (denotes as $\mathcal{W}_\delta$) whose distance to Δ_c is less than R, and task similarity ρ is less than t. Then , S_1 finds N_s skyline workers from $\mathcal{W}_\delta$ to execute the sensing task. After the task is finished, the users are required to submit an evaluation form with d reviewing questions to the S_1, which contains their opinions for evaluating workers' performance. At the end of the day, after all the tasks are finished, S_1 and S_2 updates every worker's *PS-score* by the reviews in an offline manner.

Worker's location information. In the experiment, each worker is randomly assign a pair of coordinates (x_w, y_w) such that $x_w, y_w \in (0, 2000m)$.

Worker's task similarity. For each MCS task, a worker's task similarity ρ is set to be a random value $\in (0, 1)$. We assume that the smaller the ρ is, the higher similarity between the worker's activity and the announced sensing task.

Worker's asking price. We assume that a worker's asking price is always proportional to the distance between his/her location Δ_w and the task's location Δ_c. The following equation is used to normalize each worker's asking price: $\mu = Dis(w, \Delta_c)/Dis_{max}$, where $Dis(w, \Delta_c) = \sqrt{(x_w - x_c)^2 + (y_w - y_c)^2}$, and Dis_{max} is the possible maximum

Table 4.6: The summary of simulation settings

Parameters	Settings
Simulation area	$\mathcal{A} = 2000m \times 2000m$
Worker's parameters	
The number of workers	l = 40, 60, 80, 100
Workers' locations	$x_w, y_w \in [0, 2000m]$
Task similarity	$\rho \in [0, 1]$
Asking price	$\mu = Dis(w, \Delta_c)/Dis_{max}$
Probabilistic skyline value	The ranking of *PS-score*
Process parameters	
Location filtering threshold	$R \in [300m, 800m]$
Task similarity filtering threshold	$t = 0.40, 0.42, 0.44, 0.46$
The number of skyline workers	$N_s = 1, 2, 3, 4$
The number of reviews for workers	v froms normal distribution
The number of questions in reviews	d = 5, 10, 15, 20

distance between Δ_w and Δ_c.

Worker's PS-score. According to the definition of probabilistic skyline in Section 3.1.3, each worker's *PS-score* is a value between 0 and 1. However, for simplicity, we use the ranking score of each worker's *PS-score* instead of their real values in our experiment. For instance, if there are 100 workers, then each worker's *PS-score* is assigned to be a random value between 0 and 1. After that, all the workers are sorted decreasingly by *PS-score*, and each worker's *PS-score* is reassigned by his/her own ranking, i.e., a worker's *PS-score* is set to be 1 (100) if this worker has the largest (smallest) *PS-score* value. A smaller *PS-score* means higher probabilistic skyline values.

The detailed parameter settings in the simulations are summarized in Table 4.6. We perform the experiments with Java programming language and conduct experiments on an Intel(R) Core(TM) i7-6700 CPU @3.40GHz Windows System with 32GB RAM. In order to more accurately evaluate the running time, the average results are reported. Specifically, the simulations are repeated 10 times for updating all the workers' *PS-score*; the rest of the experiment runs 10000 times.

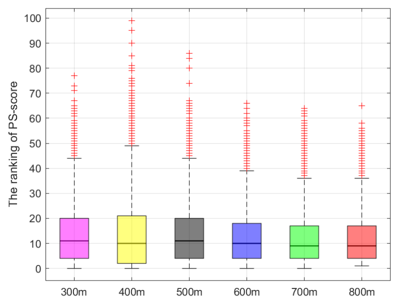
4.5.2.2 Simulation Results

Fig. 4.5 shows the number of candidate workers in different experimental settings, i.e., numbers of workers l = 40, 60, 80, 100, location filtering threshold R = $500m, 600m, 700m, 800m$, and similarity task filtering threshold t = 0.40, 0.42, 0.44, 0.46. From Fig. 4.5, we can see that the number of candidate workers is proportional to both l, t and R. The total number of active workers in a specific targeted sensing area during a certain period time is always fixed, so it is significant to set proper thresholds to get a reasonable number of matching candidates. The size of candidate workers can bring positive effect on the reliability of the selected skyline workers. The MCS platform can avoid randomness and uncertainty if more candidates are taken into consideration. Especially when the number of total workers in $\mathcal{A}$ is small, it is better to set relatively larger filtering thresholds to get sufficient candidates.

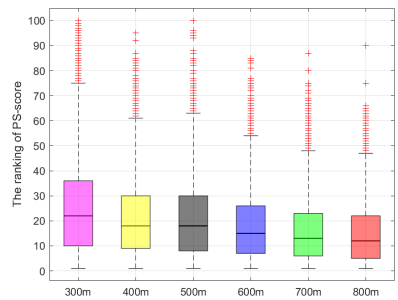
Fig. 4.6 compares the running time for selecting different numbers of skyline workers with different parameter settings. In this experiment, N_s = 1, 2, 3, or 4, l is fixed to 100, t is set to 0.46, R varies from $500m$ to $800m$. (1) We observe that for each scenario, worker selection can be finished in a very short period of time (in the order of 1×10^{-4} seconds), which validates the efficiency of our proposed scheme. (2) When N_s = 1, the running time is the lowest for all the cases. In our proposed scheme, the sum of attributes $S(w_i)$ for all the workers in $\mathcal{W}_\delta$ are calculated at first. Then the worker with the minimum $S(w_i)$ is added in the skyline pool as the first selected worker (see in Algorithm 2). This algorithm ensures that the worker with the lowest $S(w_i)$ value is the skyline worker without comparing the dominance relationship. As expected, Fig. 4.6 shows that the running time is the lowest among different experimental settings when $N_s = 1$. (3) We can see that when R is small, the running time for selecting skyline workers tends to be longer. According to Algorithm 2, the running time for skyline worker selection highly depends on the number of dominance-relationship comparisons between two workers. After all the candidate workers are being sorted by their $S(w_i)$, the 1st worker with minimum $S(w_i)$ will be chosen as the skyline worker. When N_s is larger than 1, the rest of the candidate workers need to be compared with the skyline workers, one worker will be selected as a new skyline worker if he/she is not dominated by any existing skyline worker. Otherwise, the next worker in the ranking list will be compared; this process will be repeated until all the N_s skyline workers are found. When R is large, the number of candidate workers is large (see in Fig. 4.5), and the workers who are sorted in the top positions by $S(w_i)$ are more likely to be the skyline workers. This explains why the running time for skyline worker selection tends to be longer when R is small.

Fig. 4.7 depicts the running time for updating *PS-score* for all the workers with different parameters. In the experiment, l = 40, 60, 80, or 100, the average number of reviews for each worker v is from a normal distribution with mean = 4, 6, or 8 and the standard deviation = 1, and the number of questions d = 5, 10, 15, or 20. From Fig. 4.7, we can see that the running time for updating *PS-score* is proportional to both l, v, and d. In Fig. 4.7, when $l = 100, v = 8$, and $d = 20$, the maximum running time is about 10×10^4 seconds (around 27 hours), which is quite large. However, the result is obtained by a local computer with a single processor. The fast evolution of parallel and distributed computing can provide more efficient solutions to address running time issues. In a real-world MCS platform, this computation can be distributed over a cluster of online computing machines, and the running time might be decreased to a reasonable level.

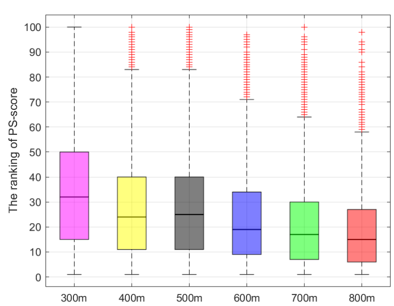
In Fig. 4.8, using the ranking of *PS-score* as the evaluation metric, we compare the reliability of selected skyline workers. In the experiment, we set $l = 100$, $N_s = 4$, $t = 0.46$, and R varying from $300m$ to $800m$. It is worth to note that the lower ranking score indicates the higher trustability of a worker. From Fig. 4.8, we can



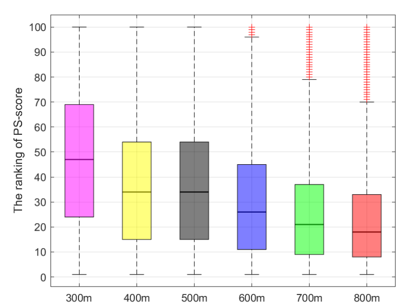
(a) The ranking of *PS-score* for the 1st skyline worker when the total number of workers l is set to 100, the task similarity threshold t is set to 0.46, the number of selected skyline workers N_s is set to 4, and the location filtering threshold R varies from $300m$ to $800m$



(b) The ranking of *PS-score* for the 2nd skyline worker when the total number of workers l is set to 100, the task similarity threshold t is set to 0.46, the number of selected skyline workers N_s is set to 4, and the location filtering threshold R varies from $300m$ to $800m$



(c) The ranking of *PS-score* for the 3rd skyline worker when the total number of workers l is set to 100, the task similarity threshold t is set to 0.46, the number of selected skyline workers N_s is set to 4, and the location filtering threshold R varies from $300m$ to $800m$



(d) The ranking of *PS-score* for the 4th skyline worker when the total number of workers l is set to 100, the task similarity threshold t is set to 0.46, the number of selected skyline workers N_s is set to 4, and the location filtering threshold R varies from $300m$ to $800m$

Figure 4.8: The ranking of *PS-score* of selected skyline workers varies with different experiment settings

see that the rankings of the first selected skyline workers are the lowest (around 10) and are similar among different location filtering thresholds. In addition, under the same R, the trustability of the selected four skyline workers can be sorted as: the 1st worker > the 2nd worker > the 3rd worker > the 4th worker. This result validates the observation in Fig. 4.6 that the workers who are sorted in the top positions of the ranking list show higher trustability than the latter ones. Finally, we can see that the skyline workers from a larger size of candidates (i.e., larger R) are more reliable than those from a smaller size of candidates (i.e., smaller R). In the real-world applications, in order to select reliable and trustable workers, it is better to generate sufficient candidate workers.

4.6 Related Work

In this section, we briefly review some related works which target the worker selection in MCS. Ren et al. [90] introduced a reputation management scheme for selecting the well-suited workers under a fixed task budget; however, they do not consider the potential privacy leakage for the workers. Kazemi and Shahabi [48] studied the problem of spatial task assignment for spatial crowdsourcing. To maximize the overall number of assigned tasks, their scheme highly depends on the locations of mobile workers, in which workers' essential personal information might be disclosed. Xiong et al. [118] investigated the problem of energy-efficient task allocation. Their research aims to guarantee a minimum number of anonymous workers who return their sensing results within a specified time-frame. Kazemi et al. [49] proposed a framework to evaluate the validity of the results provided by workers with different trust levels in MCS. They assume that every worker has a reputation score that states the trustability of this worker. However, in their study, they do not describe how the reputation score is calculated. Recently, more studies focus on the problem of privacy-enhanced worker selection. Ni et al. [75] proposed a privacy-preserving MCS worker selection framework based on the points of interest and the location of users. Jin et al. [45] present an incentive mechanism for worker selection in MCS services, which can choose workers that are more likely to provide reliable data, while protecting workers' privacy as well. Guo et al. [35] designed a MCS scheme for multi-task worker selection, which considers both workers' intentional movement for time-sensitive tasks and unintentional movement for delay-tolerant tasks. Wang et al. [111] provided a personalized privacy-preserving task allocation framework for MCS applications.

Different from the above, our work applies the (probabilistic) skyline computation to select suitable workers securely. The proposed scheme can solve the fundamental problem of how workers' trustability scores are calculated while guaranteeing the privacy of their personal information.

4.7 Concluding Remarks

In this chapter, we have proposed a privacy-preserving worker selection scheme for MCS applications based on (probabilistic) skyline computation. Our proposed scheme can efficiently select suitable and reliable workers without revealing workers' personal information. Specifically, the privacy-preserving probabilistic skyline is used to calculate workers' *PS-score* based on historical reviews. *PS-score* reveals the fluctuation of workers' past performance and can be used as their trustability. Then, skyline computation is used for selecting workers in terms of ask price, task similarity, and trustability. A non-interactive encrypted integer comparison protocol called *NIEC* is designed for supporting our scheme. Security analysis demonstrates that our scheme is privacy-preserving, theoretical performance evaluation verifies the efficiency of the proposed scheme in terms of storage and computational costs. Besides, extensive simulations have been conducted to demonstrate the effectiveness of the proposed scheme. For future work, the probabilistic skyline over sliding windows will be studied to enhance the performance of the current scheme.

Chapter 5

Continuous Probabilistic Skyline Query for Secure Worker Selection in MCS

5.1 Introduction

In the previous chapter, we proposed a novel privacy-preserving worker selection scheme based on (probabilistic) skyline computation in MCS applications, which can not only calculate workers' trustability based on historical reviews, but also select reliable workers without revealing their sensitive information. However, instead of considering the dynamic situations, our approach treats MCS as a short-term and static process. In many real-world problems, a sensing task is required to be executed continuously over a long-time period. For example, a traffic monitoring platform may observe and detect car speeds for weeks; an environment evaluation system may keep track of the local air quality for months. Unlike a static system, such sensing problems need to select proper workers repeatedly in a continuous manner. Moreover, since the workers can continuously arrive at/leave the system, the stream of sensing data is always time-sensitive. The platform is generally more interested in the recent data than those in the far past. Nevertheless, most of the studies only take workers' non-temporal characteristics (e.g., asking price and real-time location) into account. The recency and time-sensitivity of the mobile workers have not been adequately studied in the past.

Aiming to address the issues above, in this chapter, we propose an efficient and privacy-preserving worker selection scheme for MCS applications. More specifically, a probabilistic skyline based approach is proposed for continuously selecting workers over sliding windows. Furthermore, an ElGamal encryption-based scheme is designed for securely outsourcing and comparing workers' sensitive information. Overall, the main contributions of this work are three-fold as follows.

- First, we devote our attention to the problem of privacy-preserving and continuous worker selection in MCS services. By deploying probabilistic skyline queries over sliding windows, our approach can select qualified workers for each current window in terms of working experience, expiry time, and trustability. The MCS system can be kept reliable and sustainable in different application scenarios.

- Second, we design novel encryption schemes for securely outsourcing and comparing workers' sensitive attributes (i.e., working experience and trustability) without disclosing their real values. These schemes are later used for confidentially determining workers' skyline dominance relationships and calculating workers' probabilistic skyline values in worker selection.

- Third, we analyze the security of the proposed scheme and show that it can effectively preserve the privacy of workers' personal data. Besides, extensive performance evaluations are conducted on both real-world and simulated datasets, and the results demonstrate that our approach outperforms the baseline method in different application scenarios.

The remainder of this chapter is organized as follows. In Section 5.2, we introduce our system model, security model and design goals. In Section 5.3, we present the proposed scheme in details. Then in Section 5.4 and Section 5.5, we respectively present the security analysis and performance evaluation, followed by the related works in Section 5.6. Finally, we recap the conclusions in Section 5.7.

5.2 Models and Design Goals

In this section, we formalize our system model, security model, and identify our design goals. The frequently used notations are listed in Table 5.1.

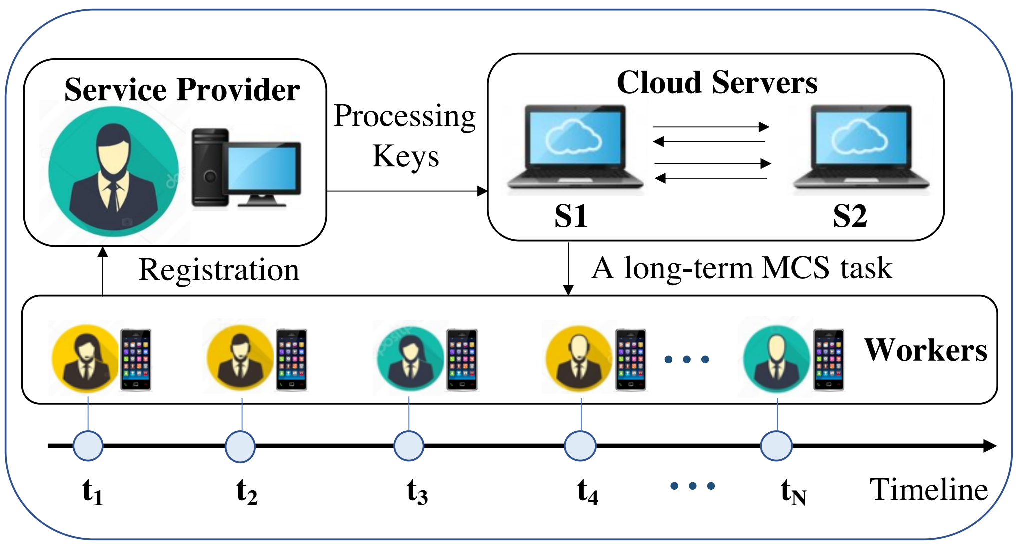


Figure 5.1: The system model under consideration in chapter 5.

5.2.1 System Model

In our system model, a long-term MCS task is denoted as $\mathcal{M}$, where workers can continuously arrive at/leave the MCS system. More specifically, we design an MCS worker selection scheme based on continuous probabilistic skyline computation. Our system model consists of four entities, namely a service provider $\mathcal{SP}$, two cloud servers $(\mathcal{S}_1, \mathcal{S}_2)$, and a set of participating workers $\mathcal{W} = \{w_1, w_2, \cdots\}$, which are shown in Fig. 5.1.

• Service Provider $(\mathcal{SP})$: $\mathcal{SP}$ is the service organizer and provider, and is responsible for bootstrapping the entire system. $\mathcal{SP}$ generates and distributes proper keys to different authorized entities so that a certain task can be completed cooperatively. Besides, $\mathcal{SP}$ assigns and computes a trustability score T_i to every registered worker w_i, which should be securely outsourced to $\mathcal{S}_1$ and $\mathcal{S}_2$ before a sensing task starts. After the task finishes, $\mathcal{SP}$ updates workers' T_i based on their sensing performance by the method in [136]. We assume that all parties fully trust $\mathcal{SP}$ in the system.

• Cloud Servers $(\mathcal{S}_1, \mathcal{S}_2)$: There are two cloud servers in our system model. After $\mathcal{M}$ starts, $\mathcal{S}_1$ and $\mathcal{S}_2$ will work together to select reliable workers over sliding windows. Concretely, when a new worker arrives, $\mathcal{S}_1$ and $\mathcal{S}_2$ need to update the subset of suitable workers for the current window based on their probabilistic skyline values.

• Workers $\mathcal{W} = \{w_1, w_2, \cdots\}$: Workers are the participants who wish to conduct $\mathcal{M}$. In our system model, each worker w_i is associated with a trustable score $T_i \in (0, 1)$, which can be considered as the trustable level of w_i. Higher T_i means w_i is more reliable for collecting and sharing high-quality sensing data. Once w_i plans to conduct $\mathcal{M}$, she/he is required to submit some real-time information to the cloud servers, e.g., arrival time $t_i^{\mathtt{arr}}$, the life span of the sensing data $t_i^{\mathtt{ls}}$, and working experience $\mathcal{E}_i$. w_i's expiry time $t_i^{\mathtt{exp}}$ can be calculated as $t_i^{\mathtt{exp}} = t_i^{\mathtt{arr}} + t_i^{\mathtt{ls}}$. After that, the cloud servers can select the suitable workers based on these information. In this work, $\mathcal{E}_i$ is defined as an integer within certain range (e.g., [1, 30]) to reveal how often w_i is allocated to conduct similar tasks in the past and the larger value is more preferred.

5.2.2 Security Model

In our security model, we consider $\mathcal{SP}$ is trustable, while $\mathcal{S}_1$ and $\mathcal{S}_2$ are honest-but-curious, which means both of them strictly follow the protocol procedure, yet may be curious to learn additional personal information in the process of worker selection. In addition, there is no collusion between $\mathcal{S}_1$ and $\mathcal{S}_2$. For the workers, we assume that they are strategic and selfish for maximizing their profits. Specifically, our model selects workers in terms of working experience, expiry time, and trustability. Therefore, if a worker w_i provides dishonest personal information (e.g., a larger $\mathcal{E}_i$ or $t_i^{\mathtt{ls}}$), she/he may have a higher probability to be selected. However, w_i' trustability T_i will be evaluated and updated based on his/her sensing performance after each task finishes [136]. So the value of T_i will be decreased largely if w_i submits false information, and the sensing results do not match the provided information.

Moreover, each worker ID can only be registered once, so malicious attackers cannot participate in $\mathcal{M}$ with multiple IDs. As a result, for long-term considerations, in order to be selected again in the future, the workers need to provide their personal information as correct as possible. It is worth noting that there may exist outside attackers who want to exploit the vulnerability of the platform and try to monitor and modify the sensing data, but they are beyond the scope of this work, and will be discussed in our future work.

5.2.3 Design Goals

This work aims to securely select a subset of reliable and trustable workers over sliding windows for conducting the MCS tasks. Specifically, the following two objectives should be satisfied.

• *Privacy preservation:* Our study needs to consider the leakage of workers' privacy since lots of sensitive information can be involved and inferred in the designed model. For example, a worker w_i's working experience $\mathcal{E}_i$ may indicate his/her daily routine

Table 5.1: The summary of important notations used in chapter 5

Notation	Definition
System Notations	
$\mathcal{SP}$	Service provider
$(\mathcal{S}_1, \mathcal{S}_2)$	Two cloud severs
$\mathcal{M}$	A long-term MCS Task
Worker Notations	
$\mathcal{W} = \{w_1, w_2, \cdots\}$	A set of registered workers
$\mathcal{W}_c = \{w_1, \cdots, w_{win}\}$	The N_{win} workers in current window
w_i	The i^{th} worker
N_i	# of workers who do not dominate w_i
T_i	Trustability of worker w_i
$\overline{T}_i$	$1 - T_i$
$t_i^{\texttt{arr}}$	Arrival time of w_i
$t_i^{\texttt{ls}}$	Life span of w_i's sensing data
$t_i^{\texttt{exp}}$	Expiry time of w_i's sensing data
$\Pr^{\texttt{sky}}(w_i)$	Probabilistic skyline of worker w_i
Worker Selection Notations	
N_{win}	The size of each sliding window
$S_{\mathcal{E}}$	Sum of probabilistic work experience
$t_i^{\texttt{rexp}}$	Relative departure time of w_i
$S_{\mathcal{E}}^{\texttt{r}}$	$S_{\mathcal{E}} - \mathcal{E}_i$ when w_i leaves the system

and behavior, which should be highly protected. Moreover, w_i's trustability T_i can be used by MCS opponents for unfair competition. So, neither $\mathcal{S}_1$ nor $\mathcal{S}_2$ can get access to the real values of $\mathcal{E}_i$ and T_i during the process of worker selection.

• *Efficiency:* In order to preserve workers' privacy, certain secure comparison protocols should be designed for continuous skyline computation. Processing the encrypted data between cloud servers will bring extra computational cost, which should be minimized in our proposed scheme. More specifically, numerous sensing data can speedily arrive in our system model. In order to find the most qualified workers for executing the MCS task in a real-time fashion, the worker selection scheme should be performed efficiently without considering network delays.

5.3 The Proposed Scheme

In this section, we present the probabilistic skyline based scheme for continuous and secure worker selection in MCS applications. The proposed scheme mainly consists of the following five phases, namely, System Preparation (`SysPre`), Outsourcing Workers' Information (`OutInf`), Comparing Workers' Working Experience (`CmpExp`), Comparing Workers' Probabilistic Skyline Values (`CmpSky`), and Worker Selection (`WrkSel`). First, in `SysPre` phase, $\mathcal{SP}$ performs some preparations for the sensing task $\mathcal{M}$, including registering and authenticating workers, generating and distributing proper keys to the correct entities, outsourcing workers' trustability to the cloud servers, and releasing the details of the sensing task $\mathcal{M}$ to both $(\mathcal{S}_1, \mathcal{S}_2)$ and $\mathcal{W}$. Once a registered worker w_i wants to participate in $\mathcal{M}$, he/she needs to outsource his/her working experience T_i, arrival time $t_i^{\texttt{arr}}$, and the life span of the sensing data $t_i^{\texttt{ls}}$ to the cloud servers in `OutInf` phase. Upon receiving workers' information, the cloud servers needs to compare the relation of their working experience and determine their skyline dominance relationship in `CompExp` phase. Consequently, the cloud servers can compare workers' probabilistic skyline values in `CmpSky` phase and finally select the suitable workers for conducting the sensing task $\mathcal{M}$ in `WrkSel` phase. The details of the five phases are introduced as follows.

5.3.1 The `SysPre` Phase

In this phase, $\mathcal{SP}$ will run the following procedures for preparing and initializing the system before the sensing task $\mathcal{M}$ starts.

• *Registration:* Once a worker w_i with identity $\texttt{ID}_i$ wants to register him/herself to the system, $\mathcal{SP}$ first validates the authenticity of $\texttt{ID}_i$ and check whether this $\texttt{ID}_i$ has been previously registered or not. If the $\texttt{ID}_i$ is authentic and valid (i.e., it has not been resigstered before), then $\mathcal{SP}$ uses a cryptographic hash function to compute a pseudo-id $\texttt{PID}_i$ based on $\texttt{ID}_i$ [65]. Moreover, w_i's trustability T_i is initialized as 0.5. For $\mathcal{S}_1$ and $\mathcal{S}_2$, $\mathcal{SP}$ needs to authenticate their identities as well before $\mathcal{M}$ starts. Once w_i starts sensing the data, the pseudo-id $\texttt{PID}_i$ needs to be authenticated again to guarantee that only registered workers can participate in $\mathcal{M}$. After $\mathcal{M}$ finishes, $\mathcal{SP}$ updates each T_i in an offline manner based on worker w_i's sensing performance [136]. The updated T_i will be used for worker selection in the next sensing task.

• *Key Generation and Distribution:* $\mathcal{SP}$ generates a key pair $(pk_{s_1}, sk_{s_1}) = ((p, q, g, y), x)$ of ElGamal encryption. Specifically, p and q are large primes and $q|(p-1)$. Let $\mathbb{Z}_p^* = \{1, 2, ..., p-1\}$, and $\mathbb{Z}_q^* = \{1, 2, ..., q-1\}$, g is a generator of a subgroup with order q in $\mathbb{Z}_p^*$, $y = g^x \bmod p$ where x is a random number from $\mathbb{Z}_q^*$. Moreover, $\mathcal{SP}$ selects another generator $\tilde{g}$ such that $\tilde{g} \neq g$, and creates a bloom filter BF based on Algorithm 6. After that, pk_{s_1} is published as system parameter to all the entities, and $\mathcal{SP}$ securely sends (sk_{s_1}, BF) and $\tilde{g}$ to $\mathcal{S}_1$ and $\mathcal{W}$, respectively.

Algorithm 6 Generation of a bloom filter

Input: A N-bit length array $A[N]$ where all the bits are initialized to 0, k independent hash functions $\mathcal{H} = \{H_1, H_2, ..., H_k\}$ where $H_n : \{0,1\}^* \rightarrow \{0, 1, ..., N-1\}$ for $n \in [1, k]$, and $\mathcal{E}_{max}$ which indicates the maximum value of working experience (e.g., $\mathcal{E}_{max} = 30$).
Output: The bloom filter BF that contains all the elements of $o = \tilde{g}^{i-j} \bmod p$ for $i, j \in [1, \mathcal{E}_{max}]$ and $i > j$

```
for i = 1 to E_max do
   for j = 1 to E_max do
      if i > j then
         o = g̃^(i−j) mod p
         for l = 1 to k do
            set A[H_l(o)] = 1
         end for
      end if
   end for
end for
return  The bloom filter BF
```

• *Outsourcing Workers' Trustability to $\mathcal{S}_1$ and $\mathcal{S}_2$:* As we mentioned before, each worker w_i's T_i and $\overline{T}_i$ are owned by $\mathcal{SP}$, and both are essential for worker selection.

In this work, we design a privacy-preserving scheme based on ElGamal encryption, which enables $\mathcal{SP}$ to securely outsource T_i and $\overline{T}_i$ to the cloud servers. The details of the proposed scheme are described as follows.

Step 1. T_i and $\overline{T}_i$ are both in the range of (0, 1) where the modulo operations can not be adopted (e.g., $T_i = 0.6755$, and $\overline{T}_i = 0.3245$). So at the beginning, $\mathcal{SP}$ multiplies both T_i and $\overline{T}_i$ by a large integer θ (e.g., $\theta = 10^4$). After that, we can get $T_i = T_i \cdot \theta = 6755$ and $\overline{T}_i = \overline{T}_i \cdot \theta = 3245$, such that the modulo operations can be applied in the further steps. It is worth noting that after expansion, $T_i + \overline{T}_i = 10^4$ rather than 1. So the Eq. (3.5) will be modified as:

$$\mathrm{Pr}^{\mathsf{sky}}(w_i) = T_i \cdot \overline{\mathrm{Pr}}(w_i) = T_i \cdot \prod_{\substack{w_k \in \mathcal{W}_c, \\ i \neq k, w_k \prec w_i}} \overline{T}_k \cdot 10^{4N_i}. \tag{5.1}$$

Step 2. $\mathcal{SP}$ selects two random numbers $r_i, \tilde{r}_i \in \mathbb{Z}_q$, and then encrypts T_i and $\overline{T}_i$ as follows:

$$\begin{aligned} T'_{i1} &= T_i \cdot y^{r_i} \bmod p, \quad \textit{and} \;\; T'_{i2} = g^{r_i} \bmod p \\ \overline{T}'_{i1} &= \overline{T}_i \cdot y^{\tilde{r}_i} \bmod p, \quad \textit{and} \;\; \overline{T}'_{i2} = g^{\tilde{r}_i} \bmod p \end{aligned} \tag{5.2}$$

Step 3. $\mathcal{SP}$ securely distributes $(T'_{i1}, \overline{T}'_{i1})$ and $(T'_{i2}, \overline{T}'_{i2})$ to $\mathcal{S}_2$ and $\mathcal{S}_1$, respectively.

• *Launching the Sensing Task* $\mathcal{M}$: After outsourcing workers' trustability, $\mathcal{SP}$ releases the detailed information of $\mathcal{M}$ to $\mathcal{S}_1, \mathcal{S}_2$, and $\mathcal{W}$, e.g., the task content, location, and starting/ending time. Moreover, we assume that only a small number of workers are useful and can be kept in each sliding window. So, $\mathcal{SP}$ defines N_{win} as the size of each sliding window and announces N_{win} to $\mathcal{S}_1$ and $\mathcal{S}_2$ as well. In `WrkSel` phase, at most N_{win} workers can be selected to fulfill $\mathcal{M}$ in each window. Usually, N_{win} is a small integer (e.g., 5 or 10), which can be determined based on previous task experience.

5.3.2 The `OutInf` Phase

After $\mathcal{M}$ starts, any worker w_i who wants to participate in $\mathcal{M}$ needs to send his/her working experience $\mathcal{E}_i$, arrival time $t_i^{\texttt{arr}}$, life span of the sensing data $t_i^{\texttt{ls}}$ to the cloud servers. As mentioned before, $\mathcal{E}_i$ is sensitive and should be protected from being disclosed. The following details describe how w_i outsources $\mathcal{E}_i$, $t_i^{\texttt{arr}}$, and $t_i^{\texttt{ls}}$ to $\mathcal{S}_1$ and $\mathcal{S}_2$.

Step 1. At first, w_i generates a random number $\bar{r}_i \in \mathbb{Z}_q$, and computes $\mathcal{E}'_{i1}$ and $\mathcal{E}'_{i2}$ as follows:

$$\mathcal{E}'_{i1} = \tilde{g}^{\mathcal{E}_i} \cdot y^{\bar{r}_i} \bmod p, \ \ and \ \ \mathcal{E}'_{i2} = g^{\bar{r}_i} \bmod p \tag{5.3}$$

Step 2. Then, w_i calculates expiry time $t_i^{\texttt{exp}}$ by $t_i^{\texttt{exp}} = t_i^{\texttt{arr}} + t_i^{\texttt{ls}}$, where $t_i^{\texttt{arr}}$ indicates his/her arrival time, and $t_i^{\texttt{ls}}$ indicates the how long he/she plans to stay in the system.

Step 3. Finally, w_i securely distributes ($\mathcal{E}'_{i1}$, $t_i^{\texttt{exp}}$) and $\mathcal{E}'_{i2}$ to $\mathcal{S}_2$ and $\mathcal{S}_1$, respectively.

5.3.3 The `CmpExp` Phase

In this phase, for w_i and w_j, $\mathcal{S}_2$ and $\mathcal{S}_1$ need to compare the relation of their working experience and determine their skyline dominance relationship.

• *Comparing workers' working experience*: Both $\mathcal{E}_i$ and $\mathcal{E}_j$ are sensitive information and should be protected from being disclosed. The following steps are conducted by $\mathcal{S}_1$ and $\mathcal{S}_2$ for comparing $\mathcal{E}_i$ and $\mathcal{E}_j$ without revealing their real values.

Step 1. Given $\mathcal{E}'_{i1}$ and $\mathcal{E}'_{j1}$ respectively from w_i and w_j, $\mathcal{S}_2$ calculates C_1 by Eq. (5.4) and then sends C_1 to $\mathcal{S}_1$.

$$C_1 = \frac{\mathcal{E}'_{i1}}{\mathcal{E}'_{j1}} = \frac{\tilde{g}^{\mathcal{E}_i} \cdot y^{\bar{r}_i}}{\tilde{g}^{\mathcal{E}_j} \cdot y^{\bar{r}_j}} = \tilde{g}^{(\mathcal{E}_i - \mathcal{E}_j)} \cdot y^{(\bar{r}_i - \bar{r}_j)} \bmod p \tag{5.4}$$

Step 2. Upon receiving C_1, $\mathcal{S}_1$ calculates C_2 as follows:

$$C_2 = \frac{C_1 \cdot \mathcal{E}'^{x}_{j2}}{\mathcal{E}'^{x}_{i2}} = \frac{\tilde{g}^{(\mathcal{E}_i - \mathcal{E}_j)} \cdot y^{(\bar{r_i} - \bar{r_j})}}{y^{(\bar{r_i} - \bar{r_j})}} = \tilde{g}^{(\mathcal{E}_i - \mathcal{E}_j)} \bmod p \tag{5.5}$$

Step 3. After obtaining C_2, $\mathcal{S}_1$ can determine the relationship between $\mathcal{E}_i$ and $\mathcal{E}_j$ as follows: If $C_2 = 1$, then $\mathcal{E}_i = \mathcal{E}_j$. Otherwise, $\mathcal{S}_1$ needs to check whether C_2 is in the BF or not: if yes, then $\mathcal{E}_i > \mathcal{E}_j$, if no, then $\mathcal{E}_i < \mathcal{E}_j$. After that, $\mathcal{S}_1$ securely sends the comparison result to $\mathcal{S}_2$.

The correctness of *Step 3* is as follows. If $C_2 = 1$, it is easy to have $\mathcal{E}_i - \mathcal{E}_j = 0$ and $\mathcal{E}_i = \mathcal{E}_j$. Otherwise, under the condition of $\mathcal{E}_i > \mathcal{E}_j$, we know that all the possible values of C_2 have already been added into the BF based on Algorithm 6. Therefore, by checking whether C_2 is in BF or not, $\mathcal{S}_1$ can easily know whether $\mathcal{E}_i > \mathcal{E}_j$ or $\mathcal{E}_i < \mathcal{E}_j$. Notably, since $\tilde{g}$ is only securely shared to $\mathcal{W}$ in the `SysPre` phase, $\mathcal{S}_1$ can only determine the relation between $\mathcal{E}_i$ and $\mathcal{E}_j$. Without knowing $\tilde{g}$, $\mathcal{S}_1$ has no idea on the real values of $\mathcal{E}_i$ and $\mathcal{E}_j$.

• *Comparing workers' skyline dominance relationship*: The skyline dominance relationship between each pair of workers is captured in terms of working experience and expiry time, where for these two attributes, the larger values are more preferred. Specifically for workers w_i and w_j, i) $\mathcal{S}_1$ can securely compare the relation between $\mathcal{E}_i$ and $\mathcal{E}_j$ and send the relation to $\mathcal{S}_2$, and ii) $t_i^{\mathtt{exp}}$ and $t_j^{\mathtt{exp}}$ can be readily compared by $\mathcal{S}_2$ in their plaintexts. Therefore, the skyline dominance relationship between w_i and w_j can be easily determined by $\mathcal{S}_2$ (see in Algorithm 7).

5.3.4 The `CmpSky` Phase

In this phase, $\mathcal{S}_1$ and $\mathcal{S}_2$ can compare the relation between $\Pr^{sky}(w_i)$ and $\Pr^{sky}(w_j)$ for w_i and w_j. During the comparison, their trustability should not be disclosed to other entities.

Let $\Pr^{\mathtt{sky}}_{max}$ denote the possible largest value for $\Pr^{sky}(w_i)$. Given $T_i, \overline{T}_k \in (0, 10^4)$, $\Pr^{\mathtt{sky}}_{max} = 10^4 \cdot \prod_1^{N_{win}} 10^4 = 10^{4(N_{win}+1)}$, where N_{win} is the window size. In our system

Algorithm 7 Workers' skyline dominance relationship comparison *WorkerDom*(w_i, w_j)

Input: $t_i^{\texttt{exp}}, t_j^{\texttt{exp}}, BF, C_2$
Output: The skyline dominance relationship between w_i and w_j;
if C_2 *in* BF **then**
    if $t_i^{\texttt{exp}} \geq t_j^{\texttt{exp}}$ **then**
        return $w_i \prec w_j$
    end if
    if $t_i^{\texttt{exp}} < t_j^{\texttt{exp}}$ **then**
        return $w_i \not\prec w_j$ and $w_j \not\prec w_i$
    end if
end if
if C_2 *not in* BF **then**
    if $t_i^{\texttt{exp}} \leq t_j^{\texttt{exp}}$ **then**
        return $w_j \prec w_i$
    end if
    if $t_i^{\texttt{exp}} > t_j^{\texttt{exp}}$ **then**
        $w_i \not\prec w_j$ and $w_j \not\prec w_i$
    end if
end if
if $C_2 = 1$ **then**
    if $t_i^{\texttt{exp}} > t_j^{\texttt{exp}}$ **then**
        return $w_i \prec w_j$
    end if
    if $t_i^{\texttt{exp}} = t_j^{\texttt{exp}}$ **then**
        return $w_i \not\prec w_j$ and $w_j \not\prec w_i$
    end if
    if $t_i^{\texttt{exp}} < t_j^{\texttt{exp}}$ **then**
        return $w_j \prec w_i$
    end if
end if

model, the largest value for N_{win} is 20, then $\Pr^{\texttt{sky}}_{max} = 10^{84}$ and the bit length $|\Pr^{\texttt{sky}}_{max}| = 280$. Hence, $\Pr^{\texttt{sky}}(w_i)$ is a large value $\in \{0,1\}^{280}$, and the method of building a bloom filter for comparing small integers in phase `CmpExp` is computationally expensive. As a result, we design a novel ElGamal-based comparison approach for large values, which is introduced as follows.

Step 1. $\mathcal{S}_1$ generates the skyline dominance relationships for all pairs of workers in the current window, and then securely sends the relationships to $\mathcal{S}_2$.

Step 2. $\mathcal{S}_2$ first selects a random number $\alpha \in \mathbb{Z}_p$. Then, based on Eq. (5.6), $\mathcal{S}_2$ calculates C_3 and C_4 for w_i and w_j, respectively. Finally $\mathcal{S}_2$ sends $(C_3,\ C_4)$ to $\mathcal{S}_1$.

$$\begin{aligned} C_3 &= \alpha \cdot T'_{i1} \cdot \prod_{\substack{w_k \in \mathcal{W}_c \\ k \neq i, w_k \prec w_i}} \overline{T}'_{k1} \cdot 10^{4N_i} \bmod p, \\ C_4 &= \alpha \cdot T'_{j1} \cdot \prod_{\substack{w_k \in \mathcal{W}_c \\ k \neq j, w_k \prec w_j}} \overline{T}'_{k1} \cdot 10^{4N_j} \bmod p, \end{aligned} \tag{5.6}$$

where N_i and N_j represent the number of workers who do not dominate w_i and w_j, respectively.

Step 3. Upon receiving C_3, $\mathcal{S}_1$ checks the dominance relationships between w_i and other workers and find all the $\overline{T}'_{k2}$ for w_k where $w_k \prec w_i$. After that, $\mathcal{S}_1$ calculates

C_5 for w_i as:

$$\begin{aligned}
C_5 &= \frac{C_3}{(T'_{i2} \cdot \prod_{\substack{w_k \in \mathcal{W}_c \\ k \neq i, w_k \prec w_i}} \overline{T}'_{k2})^x} \bmod p \\
&= \frac{\alpha \cdot T'_{i1} \cdot \prod_{\substack{w_k \in \mathcal{W}_c \\ k \neq i, w_k \prec w_i}} \overline{T}'_{k1} \cdot 10^{4N_i}}{(g^{r_i} \cdot \prod_{\substack{w_k \in \mathcal{W}_c \\ k \neq i, w_k \prec w_i}} g^{\tilde{r}_k})^x} \bmod p \\
&= \frac{\alpha \cdot T_i \cdot y^{r_i} \cdot \prod_{\substack{w_k \in \mathcal{W}_c \\ k \neq i, w_k \prec w_i}} \overline{T}_k \cdot y^{\tilde{r}_k} \cdot 10^{4N_i}}{(y^{r_i} \cdot \prod_{\substack{w_k \in \mathcal{W}_c \\ k \neq i, w_k \prec w_i}} y^{\tilde{r}_k})} \bmod p \\
&= \alpha \cdot (T_i \cdot \prod_{\substack{w_k \in \mathcal{W}_c \\ k \neq i, w_k \prec w_i}} \overline{T}_k \cdot 10^{4N_i}) \bmod p \\
&= \alpha \cdot \mathrm{Pr}^{\mathsf{sky}}(w_i) \bmod p
\end{aligned} \tag{5.7}$$

Similarly, $\mathcal{S}_1$ calculates C_6 for worker w_j based on C_4 as:

$$\begin{aligned}
C_6 &= \frac{C_4}{(T'_{j2} \cdot \prod_{\substack{w_k \in \mathcal{W}_c \\ k \neq j, w_k \prec w_j}} \overline{T}'_{k2})^x} \bmod p \\
&= \alpha \cdot \mathrm{Pr}^{\mathsf{sky}}(w_j) \bmod p
\end{aligned} \tag{5.8}$$

After that, $\mathcal{S}_1$ generates a random number $\beta \in \mathbb{Z}_q$, calculates C_7 by Eq. (5.9), and then sends C_7 to $\mathcal{S}_2$.

$$C_7 = \alpha \cdot \beta \cdot (C_5 - C_6) = \alpha \cdot \beta \cdot (\mathrm{Pr}^{\mathsf{sky}}(w_i) - \mathrm{Pr}^{\mathsf{sky}}(w_j)) \bmod p \tag{5.9}$$

Step 4. After obtaining C_7, $\mathcal{S}_2$ calculates C_8 as follows.

$$C_8 = \frac{C_7}{\alpha} = \beta \cdot (\mathrm{Pr}^{\mathsf{sky}}(w_i) - \mathrm{Pr}^{\mathsf{sky}}(w_j)) \bmod p \tag{5.10}$$

Finally, $\mathcal{S}_2$ can determine the relation between $\mathrm{Pr}^{\mathsf{sky}}(w_i)$ and $\mathrm{Pr}^{\mathsf{sky}}(w_j)$ based on the value and bit length of C_8. Specifically, if $C_8 = 0$, then it means $\mathrm{Pr}^{\mathsf{sky}}(w_i) =$

$\Pr^{\mathsf{sky}}(w_i)$. Otherwise, if $|C_8| < \frac{|p|}{2}$ then $\Pr^{\mathsf{sky}}(w_i) > \Pr^{\mathsf{sky}}(w_j)$; if $|C_8| > \frac{|p|}{2}$ then $\Pr^{\mathsf{sky}}(w_i) < \Pr^{\mathsf{sky}}(w_j)$.

The correctness of the comparison is as follows. First, given a non-negative integer β, it is easy to know that when $C_8 = \beta \cdot (\Pr^{\mathsf{sky}}(w_i) - \Pr^{\mathsf{sky}}(w_j)) \bmod p = 0$, then $\Pr^{\mathsf{sky}}(w_i) = \Pr^{\mathsf{sky}}(w_i)$. Second, as both $\Pr^{\mathsf{sky}}(w_i)$ and $\Pr^{\mathsf{sky}}(w_j) \in (0, \Pr^{sky}_{max}]$, we know that if $\Pr^{\mathsf{sky}}(w_i) > \Pr^{\mathsf{sky}}(w_j)$, then $C_8 \in (0, \beta \cdot \Pr^{\mathsf{sky}}_{max})$. Therefore, $|C_8| < |\beta| \cdot |\Pr^{\mathsf{sky}}_{max}| = 160 + 280 = 440 < \frac{|p|}{2} = 512$. Last, if $\Pr^{\mathsf{sky}}(w_i) < \Pr^{\mathsf{sky}}(w_j)$, then $\beta \cdot (\Pr^{\mathsf{sky}}(w_i) - \Pr^{\mathsf{sky}}(w_j)) < 0$ and $C_8 = (p + \beta \cdot (\Pr^{\mathsf{sky}}(w_i) - \Pr^{\mathsf{sky}}(w_j))) \bmod p$. Here, $|C_8| \sim |p|$, hence $|C_8| > \frac{|p|}{2}$.

5.3.5 The `WrkSel` Phase

After $\mathcal{M}$ starts, workers will continuously arrive at the system. Particularly, $\mathcal{W}_c = \{w_1, w_2, ..., w_{N_{win}}\}$ denotes the N_{win} workers in current sliding window, w_n denotes the newly arrived worker, (w_i, w_j) denotes any pair of workers in $\mathcal{W}_c$, $\Pr^{\mathsf{sky}}(w_i)$ and $\Pr^{\mathsf{sky}}(w_i)_n$ denote w_i's probabilistic skyline values before and after w_n arrives, $\mathcal{T}[i][j]$ denotes the value at the ith row and the jth column in $\mathcal{T}$. The following steps are performed by $\mathcal{S}_1$ and $\mathcal{S}_2$ for worker selection.

Step 1. Before w_n arrives the platform, $\mathcal{S}_1$ and $\mathcal{S}_2$ will build a global worker selection table $\mathcal{T}$ based on Algorithm 8 (an example is shown in Table 5.2). There are three values (i.e., 0, 1, -1) in $\mathcal{T}$, which represents $\Pr^{\mathsf{sky}}(w_i) = \Pr^{\mathsf{sky}}(w_j), \Pr^{\mathsf{sky}}(w_i) > \Pr^{\mathsf{sky}}(w_j)$ and $\Pr^{\mathsf{sky}}(w_i) < \Pr^{\mathsf{sky}}(w_j)$, respectively.

Step 2. After w_n arrives, the platform needs to update the relations of probabilistic skyline among workers in $\mathcal{W}_c$ (i.e., to update the values of $\mathcal{T}[i][j]$). A naive way is to update the relations for all pairs of workers. However, it is inefficient by involving many repetitive calculations. For the workers who are not dominated by w_n, their probabilistic skyline values do not change and the comparisons between them should be avoided. To address this issue, we propose an efficient approach for updating

Algorithm 8 Generation of global worker selection table $\mathcal{T}$

Input: $\mathcal{W}_c = \{w_1, w_2, ..., w_{N_{win}}\}$
Output: A two-dimensional array $\mathcal{T}$ that indicates the realtion of workers' probabilistic skyline values
Initialize $\mathcal{T}$ to an empty 2-dimensional array with the size of $(N_{win}+1) \times (N_{win}+1)$;
for $i = 1$ *to* N_{win} **do**
  for $j = 1$ *to* N_{win} **do**
    if i = j **then**
      $\mathcal{T}[i][j] = 0$
      (indicating $\Pr^{\mathsf{sky}}(w_i) = \Pr^{\mathsf{sky}}(w_j)$)
    end if
    if $\mathcal{S}_1$ and $\mathcal{S}_2$ calculate C_8 based on phase `CmpSky` **then**
      if $C_8 = 0$ **then**
        $\mathcal{T}[i][j] = 0$
        (indicating $\Pr^{\mathsf{sky}}(w_i) = \Pr^{\mathsf{sky}}(w_j)$)
      end if
      if $C_8 < \frac{|p|}{2}$ **then**
        $\mathcal{T}[i][j] = 1$ & $\mathcal{T}[j][i] = -1$
        (indicating $\Pr^{\mathsf{sky}}(w_i) > \Pr^{\mathsf{sky}}(w_j)$)
      end if
      if $C_8 > \frac{|p|}{2}$ **then**
        $\mathcal{T}[i][j] = -1$ & $\mathcal{T}[j][i] = 1$
        (indicating $\Pr^{\mathsf{sky}}(w_i) < \Pr^{\mathsf{sky}}(w_j)$)
      end if
    end if
  end for
end for
return The global worker selection table $\mathcal{T}$

$\mathcal{T}[i][j]$ based on the following cases.

Both w_i and w_j are dominated by w_n, (i.e., $w_n \prec w_i$ and $w_n \prec w_j$), denoted as $w_n \prec (w_i, w_j)$.

Theorem 5.3.1. *If $w_n \prec (w_i, w_j)$, the relation between $Pr^{\mathsf{sky}}(w_i)$ and $Pr^{\mathsf{sky}}(w_j)$ does not change, so does the value of $\mathcal{T}[i][j]$.*

Proof. Without loss of generality, we assume that $\Pr^{\mathsf{sky}}(w_i) > \Pr^{\mathsf{sky}}(w_j)$. Given $w_n \prec (w_i, w_j)$, it is easy to obtain the following equations based on Eq. (3.5):

$$\Pr^{\mathsf{sky}}(w_i)_n = \Pr^{\mathsf{sky}}(w_i) \cdot \overline{T}_n, \quad \Pr^{\mathsf{sky}}(w_j)_n = \Pr^{\mathsf{sky}}(w_j) \cdot \overline{T}_n \tag{5.11}$$

Obviously, after w_n arrives, $\Pr^{\mathsf{sky}}(w_i)_n > \Pr^{\mathsf{sky}}(w_j)_n$ due to $\Pr^{\mathsf{sky}}(w_i) > \Pr^{\mathsf{sky}}(w_j)$. Therefore, the relation between $\Pr^{\mathsf{sky}}(w_i)$ and $\Pr^{\mathsf{sky}}(w_i)$ keeps the same, and the value of $\mathcal{T}[i][j]$ does not need to update. □

In this case, w_n does not dominate either w_i or w_j (i.e., $w_n \not\prec w_i$ and $w_n \not\prec w_j$), denoted as $w_n \not\prec (w_i, w_j)$.

Theorem 5.3.2. *If $w_n \not\prec (w_i, w_j)$, the values of $Pr^{\mathsf{sky}}(w_i)$ and $Pr^{\mathsf{sky}}(w_i)$ do not change, so does the value of $\mathcal{T}[i][j]$.*

Proof. If $w_n \not\prec w_i$, then the subset of workers who dominate w_i stays the same, and the value of the following equation $\Pr^{\mathsf{sky}}(w_i) = T_i \cdot \overline{\Pr}(w_i) = T_i \cdot \prod_{\substack{w_k \in \mathcal{W}_c, \\ i \neq k, w_k \prec w_i}} \overline{T}_k \cdot \prod_{n=0}^{N_i} 10^4 \bmod p$ does not change. The same conclusion can be made for w_j. Therefore, the values of $\Pr^{\mathsf{sky}}(w_i)$ and $\Pr^{\mathsf{sky}}(w_i)$ do not change, so does the value of $\mathcal{T}[i][j]$. □

In this case, $w_n \prec w_i$, $w_n \not\prec w_j$ and $\mathcal{T}[i][j] = -1$ (i.e., $\Pr^{\mathsf{sky}}(w_i) < \Pr^{\mathsf{sky}}(w_j)$), denoted as $w_n \otimes (w_i, w_j)$.

Theorem 5.3.3. *If $w_n \otimes (w_i, w_j)$, then the relation between $Pr^{\mathsf{sky}}(w_i)$ and $Pr^{\mathsf{sky}}(w_i)$ does not change, so does the value of $\mathcal{T}[i][j]$.*

Proof. Based on the definition of probabilistic skyline, if $w_n \otimes (w_i, w_j)$, $\Pr^{\mathsf{sky}}(w_i)_n$ and $\Pr^{\mathsf{sky}}(w_j)_n$ can be calculated as:

$$\Pr^{\mathsf{sky}}(w_i)_n = \Pr^{\mathsf{sky}}(w_i) \cdot \overline{T}_n, \quad \Pr^{\mathsf{sky}}(w_j)_n = \Pr^{\mathsf{sky}}(w_j) \cdot 10^4 \tag{5.12}$$

In this case, $\Pr^{\mathsf{sky}}(w_i) < \Pr^{\mathsf{sky}}(w_j)$, and also $\overline{T}_n < 10^4$, then it is easy to have $\Pr^{\mathsf{sky}}(w_i)_n < \Pr^{\mathsf{sky}}(w_j)_n$. Therefore, the relation between $\Pr^{\mathsf{sky}}(w_i)$ and $\Pr^{\mathsf{sky}}(w_i)$ does not change, so does the value of $\mathcal{T}[i][j]$. □

In this case, $w_n \prec w_i$, $w_n \not\prec w_j$ and $\mathcal{T}[i][j] = 1$ (i.e., $\Pr^{\mathsf{sky}}(w_i) > \Pr^{\mathsf{sky}}(w_j)$), which is denoted as $w_n \oplus (w_i, w_j)$.

Theorem 5.3.4. *If $w_n \oplus (w_i, w_j)$, then the relation between $Pr^{sky}(w_i)_n$ and $Pr^{sky}(w_j)_n$ cannot be determined without calculation. So $\mathcal{S}_1$ and $\mathcal{S}_2$ need to compare their relation based on the method in Section 5.3.4, and update $\mathcal{T}[i][j]$ accordingly.*

Proof. Similar to Case 5.3.5, given $w_n \oplus (w_i, w_j)$, $\Pr^{\mathsf{sky}}(w_i)_n$ and $\Pr^{\mathsf{sky}}(w_j)_n$ can still be calculated by Eq. (5.12). However, only given $\Pr^{\mathsf{sky}}(w_i) > \Pr^{\mathsf{sky}}(w_j)$ and $\overline{T}_n < 10^4$, we cannot determine the relation of $\Pr^{\mathsf{sky}}(w_i)_n$ and $\Pr^{\mathsf{sky}}(w_j)_n$ without updating their real values. □

In summary, after w_n arrives at the platform, $\mathcal{S}_1$ and $\mathcal{S}_2$ can update the relation between $\Pr^{\mathsf{sky}}(w_i)_n$ and $\Pr^{\mathsf{sky}}(w_j)_n$ (i.e., the values of $\mathcal{T}[i][j]$) according to the cases mentioned above.

Step 3. Next, $\mathcal{S}_1$ and $\mathcal{S}_2$ compare the relations of probabilistic skyline values between w_n and all the workers in $\mathcal{W}_c$, and then fill in $\mathcal{T}$. For example, we assume $\mathcal{W}_c = \{w_1, w_2, w_3, w_4, w_5\}$. If the ranking of probabilistic skyline values for all the workers is $w_5 > w_4 > w_n > w_2 > w_3 > w_1$, then after this step, $\mathcal{T}$ is shown in Table 5.2. Finally, $\mathcal{S}_2$ can select the top-N_{win} workers based on the values in $\mathcal{T}$. Specifically, $\mathcal{S}_2$ checks all the rows in $\mathcal{T}$, and finds the row which does not contain 1 (i.e., the first

Table 5.2: An example of global worker selection table $\mathcal{T}$

	w_1	w_2	w_3	w_4	w_5	w_n
w_1	0	-1	-1	-1	-1	-1
w_2	1	0	1	-1	-1	-1
w_3	1	-1	0	-1	-1	-1
w_4	1	1	1	0	-1	1
w_5	1	1	1	1	0	1
w_n	1	1	1	-1	-1	0

row in Table 5.2). This means that the worker in this row has the lowest probabilistic skyline value among all the workers (i.e., w_1 in this example). At last, w_1 will be removed from the platform, and the rest of the workers are kept for fulfilling $\mathcal{M}$.

5.4 Security Analysis

In this section, we will analyze the security properties of the proposed scheme. Notably, following the design goals illustrated in Section 5.2.3, the analysis will focus on how our scheme is privacy-preserving in protecting workers' personal information, i.e., working experience and trustability, from being disclosed in the process of worker selection. Specifically, let $\mathcal{W}_c = \{w_1, w_2, .., w_{N_{win}}\}$ denote the N_{win} workers in the current window and for $\forall w_i \in \mathcal{W}_c$, we have the following two theorems.

Theorem 5.4.1. *The worker w_i's working experience $\mathcal{E}_i$ cannot be revealed during the whole process under the assumption that $\mathcal{S}_1$ and $\mathcal{S}_2$ are honest-but-curious.*

Proof. We give the proof in four parts according to the corresponding roles in our scheme, i.e., $\mathcal{S}_1$, $\mathcal{S}_2$, other workers and outsiders.

- Under the assumption that $\mathcal{S}_1$ and $\mathcal{S}_2$ are honest-but-curious, $\mathcal{S}_1$ can only obtain $\mathcal{E}'_{i2}$ and many $(C_{1,(i,j)}, C_{2,(i,j)})$'s in our scheme, where $i, j \in [1, N_{win}]$ and

$i \neq j$. From the following equations

$$C_{1,(i,j)} = \tilde{g}^{\mathcal{E}_i - \mathcal{E}_j} \cdot g^{\bar{r}_i - \bar{r}_j} \bmod p,$$
$$C_{2,(i,j)} = \tilde{g}^{\mathcal{E}_i - \mathcal{E}_j} \bmod p, \quad \mathcal{E}'_{i2} = g^{\bar{r}_i} \bmod p,$$

we can see that $C_{2,(i,j)}$ comes from $C_{1,(i,j)}$, and the information related to $\mathcal{E}_i$ in $C_{2,(i,j)}$ is not less than that in $C_{1,(i,j)}$. Furthermore, $C_{2,(i,j)}$ is independent from $\mathcal{E}'_{i2}$ that contains nothing related to $\mathcal{E}_i$. Hence, we only need to analyze whether $\mathcal{S}_1$ can obtain $\mathcal{E}_i$ from $C_{2,(i,j)}$'s.

It is easy to see that $\mathcal{S}_1$ can obtain at most the following $N_{win} - 1$ independent equations regarding $C_{2,(i,j)}$'s.

$$\begin{cases} C_{2,(i,1)} &= \tilde{g}^{\mathcal{E}_i - \mathcal{E}_1} \bmod p \\ & \vdots \\ C_{2,(i,i-1)} &= \tilde{g}^{\mathcal{E}_i - \mathcal{E}_{i-1}} \bmod p \\ C_{2,(i,i+1)} &= \tilde{g}^{\mathcal{E}_i - \mathcal{E}_{i+1}} \bmod p \\ & \vdots \\ C_{2,(i,N_{win})} &= \tilde{g}^{\mathcal{E}_i - \mathcal{E}_{N_{win}}} \bmod p \end{cases}$$

Without loss of generality, we assume that all the worker experience values are different. Under the assumption that $\mathcal{S}_1$ is honest-but-curious, $\mathcal{S}_1$ faces $N_{win} + 1$ unknown values in the above equations. Those $N_{win} + 1$ unknown values are $(\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_{N_{win}}, \tilde{g})$. Hence, even if $\mathcal{E}_i$ has a small range, $\mathcal{S}_1$ cannot obtain any information about $\mathcal{E}_i$ from the above equations. In other words, $\mathcal{E}_i$ is kept secret from $\mathcal{S}_1$ in our scheme.

- Under the assumption that $\mathcal{S}_1$ and $\mathcal{S}_2$ are honest-but-curious, $\mathcal{S}_2$ can only obtain $\mathcal{E}'_{i1}$ that contains information about $\mathcal{E}_i$, where $\mathcal{E}'_{i1} = \tilde{g}^{\mathcal{E}_i} \cdot y^{\bar{r}_i} \bmod p$. However, $\mathcal{S}_2$ cannot obtain $\mathcal{E}_i$ only from $\mathcal{E}'_{i1}$ according to the correctness and security of ElGamal encryption. Hence, $\mathcal{E}_i$ is kept secret from $\mathcal{S}_2$ in our scheme.

- Regarding worker w_j $(i \neq j)$, he/she only knows whether him/herself is chosen or not in our scheme. w_j does not even know which worker he/she is compared with. Hence, w_j has no idea about $\mathcal{E}_i$ in our scheme.

- Regarding the ones outside of our scheme, it is clear that $\mathcal{E}_i$ is kept secret from them, since they obtain less information than $\mathcal{S}_1$, $\mathcal{S}_2$ or workers in our scheme.

□

Theorem 5.4.2. *The worker w_i's trustability T_i cannot be revealed during the whole process under the assumption that $\mathcal{S}_1$ and $\mathcal{S}_2$ are honest-but-curious.*

Proof. Given $T_i + \overline{T}_i = 10^4$, in order to protect T_i from being disclosed to other entities, our scheme also needs to guarantee that $\overline{T}_i$ is privacy-preserving in the whole process of worker selection. Specifically, the following parts are provided to prove the confidentiality of T_i to $\mathcal{S}_1$, $\mathcal{S}_2$, other workers, and outsiders.

- In the proposed scheme, $\mathcal{S}_1$ keeps both T'_{i2} and $\overline{T}'_{i2}$. According to the correctness and security of the ElGamal encryption, $\mathcal{S}_1$ can compute T_i or $\overline{T}_i$ as long as $\mathcal{S}_1$ can also obtain T'_{i1} or $\overline{T}'_{i1}$. Therefore, in order to prove this theorem, we need to demonstrate that $\mathcal{S}_1$ cannot obtain T_i, $\overline{T}_i$, T'_{i1}, and $\overline{T}'_{i1}$, respectively. The proofs are presented as follows.

  - *T_i will not be revealed to $\mathcal{S}_1$.* In our proposed scheme, the views of $\mathcal{S}_1$ that related to T_i are many $(C_{5,(i,j)}, C_{7,(i,j)})$'s, such that

$$C_{5,(i,j)} = \alpha_{(i,j)} \cdot T_i \cdot \overline{\mathbb{T}}_i \bmod p,$$
$$C_{7,(i,j)} = \alpha_{(i,j)} \cdot \beta_{(i,j)} \cdot (T_i \cdot \overline{\mathbb{T}}_i - T_j \cdot \overline{\mathbb{T}}_j) \bmod p,$$

    where $\overline{\mathbb{T}}_i$ is the continuous multiplication of $\overline{T}_k$'s and $10^{4N_i} \bmod p$ if there exists $w_k \in \mathcal{W}_c$ and $w_k \prec w_i$. If no such worker exists (w_i is a skyline worker), then $\overline{\mathbb{T}}_i = 10^{4N_i} \bmod p$, i.e.,

$$\overline{\mathbb{T}}_i = \begin{cases} \prod \overline{T}_k \cdot 10^{4N_i} \bmod p & \exists w_k \in \mathcal{W}_c, w_k \prec w_i, \\ 10^{4N_i} \bmod p & \text{otherwise.} \end{cases}$$

Based on Eq. (5.9) in Section 5.3.4, we know that $C_{7,(i,j)}$ is a linear combination of $C_{5,(i,j)}$ and $C_{6,(i,j)}$, where $C_{6,(i,j)} = \alpha_{(i,j)} \cdot \beta_{(i,j)} \cdot T_j \cdot \overline{\mathbb{T}}_j \cdot 10^{4N_j} \bmod p$. Since $C_{6,(i,j)}$ contains nothing about T_i, so the information related to T_i in $C_{7,(i,j)}$ is not more than that in $C_{5,(i,j)}$. Therefore, we only need to analyze whether $\mathcal{S}_1$ can obtain T_i from $C_{5,(i,j)}$'s or not.

It is easy to know that $\mathcal{S}_1$ can get the following $N_{win} - 1$ equations for $C_{5,(i,j)}$'s in the process of worker selection.

$$\begin{cases} C_{5,(i,1)} &= \alpha_{(i,1)} \cdot T_i \cdot \overline{\mathbb{T}}_i \bmod p \\ & \vdots \\ C_{5,(i,i-1)} &= \alpha_{(i,i-1)} \cdot T_i \cdot \overline{\mathbb{T}}_i \bmod p \\ C_{5,(i,i+1)} &= \alpha_{(i,i+1)} \cdot T_i \cdot \overline{\mathbb{T}}_i \bmod p \\ & \vdots \\ C_{5,(i,N_{win})} &= \alpha_{(i,N_{win})} \cdot T_i \cdot \overline{\mathbb{T}}_i \bmod p \end{cases}$$

If we consider $\mathbb{T}_i$ as a whole variable, then depending on whether $\mathbb{T}_i = 10^{4N_i} \bmod p$ or not, $\mathcal{S}_1$ faces at least N_{win} and at most $N_{win}+1$ unknown values in the above $N_{win} - 1$ equations. The definite N_{win} unknown values are $(\alpha_{(i,1)}, \alpha_{(i,2)}, ..., \alpha_{(i,i-1)}, \alpha_{(i,i+1)}, ...\alpha_{(i,n)}, T_i)$. Therefore, $\mathcal{S}_1$ cannot directly solve the above equations, i.e., $\mathcal{S}_1$ cannot obtain any information about T_i.

– $\overline{T}_i$ *will not be revealed to* $\mathcal{S}_1$. Let $\mathcal{W}_c^i$ denote a subset of workers in $\mathcal{W}_c$ (i.e., $\mathcal{W}_c^i \in \mathcal{W}_c$) such that every worker in $\mathcal{W}_c^i$ is dominated by w_i, so for

$\forall w_{k'} \in \mathcal{W}_c^i$, we have $w_i \prec w_{k'}$. If no such worker exists, then $\mathcal{W}_c^i = \emptyset$.

It is worth noting that if $\mathcal{W}_c^i = \emptyset$, then $\mathcal{S}_1$ can get nothing related to $\overline{T}_i$. Without lossing of generality, we assume that $\mathcal{W}_c^i \neq \emptyset$, and there is at least one worker $w_{k'}$ in $\mathcal{W}_c^i$. Accordingly, in our proposed scheme, the views of $\mathcal{S}_1$ related to $\overline{T}_i$ are many $(C_{5,(k',j)}, C_{7,(k',j)})$'s for $k' \neq j$, such that

$$C_{5,(k',j)} = \alpha_{(k',j)} \cdot T_{k'} \cdot \overline{T}_i \cdot \overline{\mathbb{T}}_{k'/i} \bmod p,$$
$$C_{7,(k',j)} = \alpha_{(k',j)} \cdot \beta_{(k',j)} \cdot (T_{k'} \cdot \overline{T}_i \cdot \overline{\mathbb{T}}_{k'/i} - T_j \cdot \overline{\mathbb{T}}_j) \bmod p,$$

where

$$\overline{\mathbb{T}}_{k'/i} = \begin{cases} \prod \overline{T}_k \cdot 10^{4N_{k'}} \bmod p & \exists w_k \in \mathcal{W}_c, \\ & w_k \prec w_{k'}, k \neq i, \\ 10^{4N_{k'}} \bmod p & \text{otherwise.} \end{cases}$$

Similarly, the information that related to $\overline{T}_i$ in $C_{7,(k',j)}$ is not more than that in $C_{5,(k',j)}$, so in the next, we only need to analyze whether $\mathcal{S}_1$ can obtain $\overline{T}_i$ from $C_{5,(k',j)}$'s or not.

It is easy to know that $\mathcal{S}_1$ can get at most the following $N_{win}-1$ equations regarding $C_{5,(k',j)}$'s in our proposed scheme.

$$\begin{cases} C_{5,(k',1)} = \alpha_{(k',1)} \cdot T_{k'} \cdot \overline{T}_i \cdot \overline{\mathbb{T}}_{k'/i} \bmod p \\ \qquad\vdots \\ C_{5,(k',k'-1)} = \alpha_{(k',k'-1)} \cdot T_{k'} \cdot \overline{T}_i \cdot \overline{\mathbb{T}}_{k'/i} \bmod p \\ C_{5,(k',k'+1)} = \alpha_{(k',k'+1)} \cdot T_{k'} \cdot \overline{T}_i \cdot \overline{\mathbb{T}}_{k'/i} \bmod p \\ \qquad\vdots \\ C_{5,(k',N_{win})} = \alpha_{(k',N_{win})} \cdot T_{k'} \cdot \overline{T}_i \cdot \overline{\mathbb{T}}_{k'/i} \bmod p \end{cases}$$

Again, let consider $\overline{\mathbb{T}}_{k'/i}$ as a whole variable. It is easy to see that $\mathcal{S}_1$

faces at least $N_{win}+1$ and at most $N_{win}+2$ unknown values, depending on whether $\mathbb{T}_{k'/i} = 10^{4N_{k'}} \bmod p$ or not. The $N_{win}+1$ definite unknown values are $(\alpha_{(k',1)}, ..., \alpha_{(k',k'-1)}, \alpha_{(k',k'+1)}, ..., \alpha_{(k',N_{win})}, T_{k'}, \overline{T}_i)$. Accordingly, $\mathcal{S}_1$ cannot obtain any information about $\overline{T}_i$ from the above equations.

– T'_{i1} *will not be revealed to* $\mathcal{S}_1$. In the proposed scheme, the views of $\mathcal{S}_1$ that related to T'_{i1} are many $C_{3,(i,j)}$'s, such that

$$C_{3,(i,j)} = \alpha_{(i,j)} \cdot T'_{i1} \cdot \overline{\mathbb{T}}'_{i1} \bmod p,$$

where $\overline{\mathbb{T}}'_{i1}$ is the continuous multiplication of $\overline{T}'_{k1}$'s and $10^{4N_i} \bmod p$ if there exists $w_k \in \mathcal{W}_c$ such that $w_k \prec w_i$. If no such worker exists (w_i is a skyline worker), then $\overline{\mathbb{T}}'_{i1} = 10^{4N_i} \bmod p$, i.e.,

$$\overline{\mathbb{T}}'_{i1} = \begin{cases} \prod \overline{T}'_{k1} \cdot 10^{4N_i} \bmod p & \exists w_k \in \mathcal{W}_c, w_k \prec w_i, \\ 10^{4N_i} \bmod p & \text{otherwise.} \end{cases}$$

In the proposed scheme, $\mathcal{S}_1$ can get at most the following $N_{win}-1$ equations regarding $C_{3,(i,j)}$'s in our proposed scheme.

$$\begin{cases} C_{3,(i,1)} & = & \alpha_{(i,1)} \cdot T'_{i1} \cdot \overline{\mathbb{T}}'_{i1} \bmod p \\ & \vdots & \\ C_{3,(i,i-1)} & = & \alpha_{(i,i-1)} \cdot T'_{i1} \cdot \overline{\mathbb{T}}'_{i1} \bmod p \\ C_{3,(i,i+1)} & = & \alpha_{(i,i+1)} \cdot T'_{i1} \cdot \overline{\mathbb{T}}'_{i1} \bmod p \\ & \vdots & \\ C_{3,(i,N_{win})} & = & \alpha_{(i,N_{win})} \cdot T'_{i1} \cdot \overline{\mathbb{T}}'_{i1} \bmod p \end{cases}$$

Similarly, $\mathcal{S}_1$ will face at least N_{win} and at most $N_{win}+1$ unknown values in the above $N_{win}-1$ equations. The definite N_{win} unknown values are

$(\alpha_{(i,1)}, ..., \alpha_{(i,i-1)}, \alpha_{(i,i+1)}, ..., \alpha_{(i,N_{win})}, T'_{i1})$. Therefore, $\mathcal{S}_1$ cannot compute T'_{i1}. Without T'_{i1}, based on the correctness and security of ElGalmal encryption, $\mathcal{S}_1$ cannot obtain T_i in the designed scheme.

– $\overline{T}'_{i1}$ *will not be revealed to* $\mathcal{S}_1$. In the proposed scheme, the view of $\mathcal{S}_1$ that related to $\overline{T}'_{i1}$ are many $C_{3,(k',j)}$, such that

$$C_{3,(k',j)} = \alpha_{(k',j)} \cdot T'_{k'1} \cdot \overline{T}'_{i1} \cdot \overline{\mathbb{T}}'_{k'1/i} \bmod p,$$

where $\overline{\mathbb{T}}'_{k'1/i}$ is the continuous multiplication of $\overline{T}'_{k1}$'s and $10^{4N_{k'}} \bmod p$ if there exists $w_k \in \mathcal{W}_c$ $(k \neq i)$ such that $w_k \prec w'_k$. If no such worker exists (i.e., $w_{k'}$ is only dominated by w_i), then $\overline{\mathbb{T}}'_{k'1/i} = 10^{4N_{k'}} \bmod p$, i.e.,

$$\overline{\mathbb{T}}'_{k'1/i} = \begin{cases} \prod \overline{T}'_{k1} \cdot 10^{4N_i} \bmod p & \exists w_k \in \mathcal{W}_c, \\ & w_k \prec w_{k'}, k \neq i, \\ 10^{4N_{k'}} \bmod p & \text{otherwise.} \end{cases}$$

Again, $\mathcal{S}_1$ can get the following $N_{win} - 1$ equations regarding $C_{3,(k',j)}$'s at most in our scheme.

$$\begin{cases} C_{3,(k',1)} = \alpha_{(k',1)} \cdot T'_{k'1} \cdot \overline{T}'_{i1} \cdot \overline{\mathbb{T}}'_{k'1/i} \bmod p \\ \qquad\vdots \\ C_{3,(k',k'-1)} = \alpha_{(k',k'-1)} \cdot T'_{k'1} \cdot \overline{T}'_{i1} \cdot \overline{\mathbb{T}}'_{k'1/i} \bmod p \\ C_{3,(k',k'+1)} = \alpha_{(k',k'+1)} \cdot T'_{k'1} \cdot \overline{T}'_{i1} \cdot \overline{\mathbb{T}}'_{k'1/i} \bmod p \\ \qquad\vdots \\ C_{3,(k',N_{win})} = \alpha_{(k',N_{win})} \cdot T'_{k'1} \cdot \overline{T}'_{i1} \cdot \overline{\mathbb{T}}'_{k'1/i} \bmod p \end{cases}$$

Let consider $\mathbb{T}'_{k'1/i}$ as a whole variable, then it is easy to know that $\mathcal{S}_1$ faces at least $N_{win}+1$ unknown variables from the above $N_{win}-1$ equations. The definite $N_{win}+1$ variables are $(\alpha_{(k',1)}, ..., \alpha_{(k',k'-1)}, \alpha_{(k',k'+1)}, ..., \alpha_{(k',N_{win})}, T'_{k'1}, \overline{T}'_{i1})$. Similar with the previous discussions, $\mathcal{S}_1$ has no idea about $\overline{T}'_{i1}$, so $\mathcal{S}_1$ cannot compute $\overline{T}_i$ based on ElGamal encryption.

In summary, we have proved that $\mathcal{S}_1$ can obtain neither T_i, $\overline{T}_i$, T'_{i1}, nor $\overline{T}'_{i1}$ in the proposed scheme. Therefore, the real value of T_i for any worker $w_i \in \mathcal{W}_c$ will not be revealed to $\mathcal{S}_1$ in the process of worker selection.

- In the proposed scheme, $\mathcal{S}_2$ keeps T'_{i1}, $\overline{T}'_{i1}$, and many $(C_{3,(i,j)}, C_{7,(i,j)})$'s. Based on the correctness and security of ElGamal encryption, $\mathcal{S}_2$ cannot compute T_i or $\overline{T}_i$ with only knowing T'_{i1} or $\overline{T}'_{i1}$. In addition, $C_{3,(i,j)}$ is computed by T'_{i1} and $\overline{T}'_{k1}$'s for $w_k \prec w_i$. So the information related to T_i or $\overline{T}_i$ in $C_{3,(i,j)}$ is no more than that in T'_{i1} and $\overline{T}'_{i1}$. Therefore, $\mathcal{S}_2$ cannot obtain T_i or $\overline{T}_i$ from either T'_{i1}, $\overline{T}'_{i1}$, or $C_{3,(i,j)}$. In the next, we will demonstrate why $\mathcal{S}_2$ cannot get access to T_i and $\overline{T}_i$ from $C_{7,(i,j)}$'s.

  It is easy to see that $\mathcal{S}_2$ can obtain the following $N_{win}-1$ equations regarding about $C_{7,(i,j)}$.

$$\begin{cases} C_{7,(i,1)} = \alpha_{(i,1)} \cdot \beta_{(i,1)} \cdot (T_i \cdot \overline{\mathbb{T}}_i - T_j \cdot \overline{\mathbb{T}}_j) \bmod p \\ \qquad\vdots \\ C_{7,(i,i-1)} = \alpha_{(i,i-1)} \cdot \beta_{(i,i-1)} \cdot (T_i \cdot \overline{\mathbb{T}}_i - T_j \cdot \overline{\mathbb{T}}_j) \bmod p \\ C_{7,(i,i+1)} = \alpha_{(i,i+1)} \cdot \beta_{(i,i+1)} \cdot (T_i \cdot \overline{\mathbb{T}}_i - T_j \cdot \overline{\mathbb{T}}_j) \bmod p \\ \qquad\vdots \\ C_{7,(i,N_{win})} = \alpha_{(i,N_{win})} \cdot \beta_{(i,N_{win})} \cdot (T_i \cdot \overline{\mathbb{T}}_i - T_j \cdot \overline{\mathbb{T}}_j) \bmod p \end{cases}$$

  Finally, $\mathcal{S}_2$ will face at least $N_{win}+1$ unknown values from the above $N_{win}-1$ equations, i.e., $(\beta_{(i,1)}, ..., \beta_{(i,i-1)}, \beta_{(i,i+1)}, ..., \beta_{(i,N_{win})}, T_i, T_j)$. Therefore, $\mathcal{S}_2$ cannot obtain T_i from $C_{7,(i,j)}$'s. In summary, T_i is privacy-preserving for $\mathcal{S}_2$ in the whole process of worker selection.

- Same as Theorem 5.4.1, it is easy to prove that neither other workers nor outsiders can obtain the information about T_i in our designed scheme.

$\square$

5.5 Performance Evaluation

In this section, we first study the performance of our proposed scheme in terms of storage overhead and computational overhead. Then, we provide a detailed description of our experimental configuration and comparison results.

5.5.1 Theoretical Analysis

• Storage Overheads: Assume there are total N workers registered in the MCS system, then in the `SysPre` phase, for each worker w_i, $\mathcal{S}_1$ needs to keep $T'_{i2}, \overline{T}'_{i2}$; $\mathcal{S}_2$ needs to store $T'_{i1}, \overline{T}'_{i1}$. After $\mathcal{M}$ starts, participating workers need to outsource $\mathcal{E}'_{i2}$ and $\mathcal{E}'_{i1}$ to $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. In summary, for both $\mathcal{S}_1$ and $\mathcal{S}_2$, the storage overheads in the process of worker selection can be computed as $\sum_1^N 2|p| + \sum_1^{N_{win}+1} |p|$, where $|p|$ (e.g., 1024) is the bit-length of large prime p generated in Section 5.3.1 and N_{win} is window size.

• Computational Overheads: The computational costs for `CmpExp` phase can be calculated as $\frac{(N_{win}+1)(N_{win}+2)}{2} \cdot t_{sky}$, where t_{sky} is the time cost for determining the skyline dominance relationship between each pair of workers. For a newly formed window with $N_{win}+1$ workers, the computational costs for `WrkSel` phase can be calculated as $N_{win} \cdot t_{w_i,w_j}$, where t_{w_i,w_j} is the time cost for comparing $\Pr^{\mathsf{sky}}(w_i)$ and $\Pr^{\mathsf{sky}}(w_j)$ in phase `CmpSky`. Therefore, the overall computational costs for worker selection is $\frac{(N_{win}+1)(N_{win}+2)}{2} \cdot t_{sky} + N_{win} \cdot t_{w_i,w_j}$.

5.5.2 Dataset, Experimental Settings, and the Baseline Method

• Real-world Dataset: The real-world dataset used in our experiment is Reality Mining Data [85]. This dataset tracks the mobile communication information of ninety-four people in the Massachusetts Institute of Technology from September 2004 to June 2005. After removing all the logs with none communication time duration, we finally extract 111,586 smartphone logs. For each log record, the time of picking up the phone represents the start time, the time duration of the communication means the life-span of the worker's sensing data. Moreover, for each worker w_i, we randomly assign $\mathcal{E}_i \in [1, 30]$ as working experience and $T_i \in (0, 1)$ as trustability, where both $\mathcal{E}_i$ and T_i follow uniform distribution.

• Synthetic Datasets: In the experiment, we also generate two different kinds of synthetic datasets (independent and anti-correlated) based on [8], both their sizes are 100k. For the independent data, each worker w_i's $\mathcal{E}_i \in [1, 30]$, $t_i^{\mathtt{ls}} \in [1s, 100s]$ and $T_i \in (0, 1)$ are generated independently using uniform distributions. For the anti-correlated data, we first uniformly generate a random probability $T_i \in (0, 1)$ as workers' trustability. Then, we assume that if a worker has higher $\mathcal{E}_i$ for a certain task $\mathcal{M}$, she/he tends to take shorter time to finish $\mathcal{M}$, i.e., smaller $t_i^{\mathtt{ls}}$. Hence, $\mathcal{E}_i$ and $t_i^{\mathtt{ls}}$ are generated using the method in [8] such that they are anti-correlated with each other. For both simulated datasets, we randomly assign an order for workers' arrival in a data stream.

• Experimental settings and the baseline method: First, we need to decide the key sizes of the proposed encryption methods, i.e., the bit length of p and q. As we mentioned before, our proposed schemes leverage ElGamal encryption, whose chosen-plaintext attack security is based on the decisional Diffie-Hellman (DDH) assumption [6]. According to [53, 30, 80], when the bit lengths of p and q are set as 1024 and 160, the DDH assumtion holds. As a result, we set $|p| = 1024$ and $|q| = 160$ in this chapter.

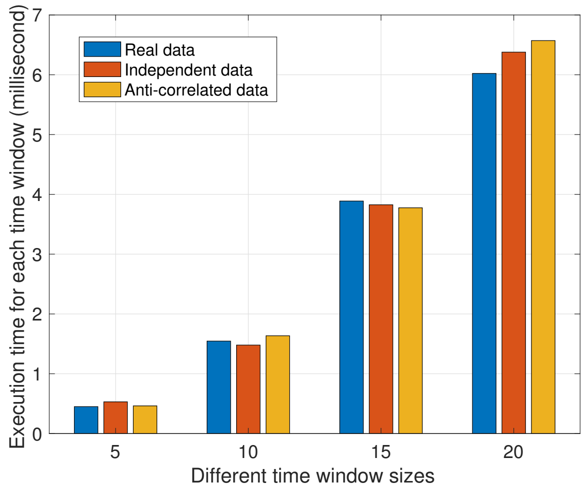
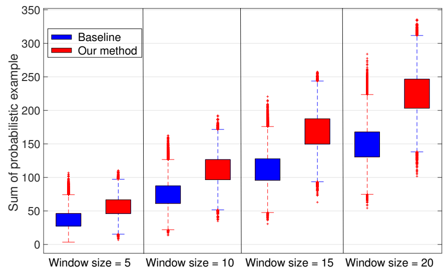


Figure 5.2: The execution time (s) of the proposed worker selection scheme for different datasets with window size N_{win} varies from 5 to 20 (the communication costs between two cloud servers $\mathcal{S}_1$ and $\mathcal{S}_2$ are not considered).

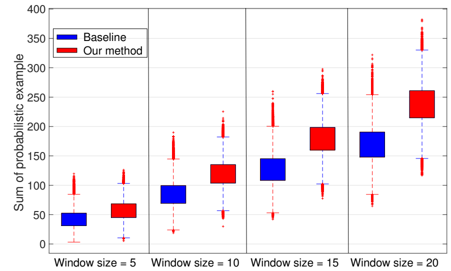
Next, we need to decide the experimental parameters for the MCS tasks. The window size N_{win} varies from 5, 10, 15, to 20, which means that our worker selection scheme will lanuch when $N_{win} + 1$ (i.e., 6, 11, 16, and 21) workers are in the system. We also proposed a baseline method for comparison purposes. More specifically, when a new worker arrives in the system, the baseline method discards the worker who has the earliest expiry time $t_i^{\texttt{exp}}$ and keeps the rest of the workers for conducting the task. We perform the experiments with Python programming language on an Intel(R) Core(TM) i7-6700 CPU @3.60GHz Windows 64-bit Operating System with 32 GB RAM. We repeated each experiment 10000 times, and the average results are reported.

5.5.3 Experimental Results

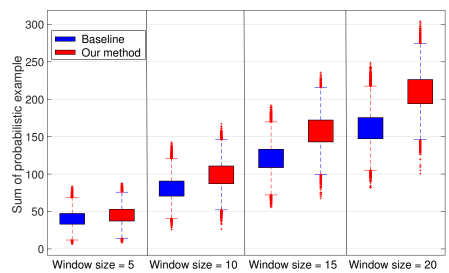
• The computational cost for worker selection in each sliding window: It is noteworthy that our proposed scheme requires constant interactions between $\mathcal{S}_1$ and $\mathcal{S}_2$. However, the communication time cost between them is hard to be simulated. So we ignored the communication cost in our experiment. From Fig. 5.2, we observe that



(a) Real dataset: Comparison of workers' quality between our method and baseline based on $S_{\mathcal{E}}$ with window size N_{win} varies from 5 to 20.



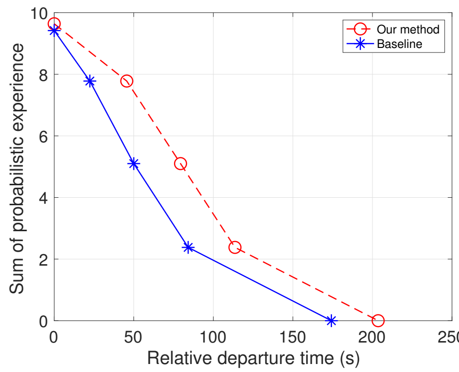
(b) Independent dataset: Comparison of workers' quality between our method and baseline based on $S_{\mathcal{E}}$ with window size N_{win} varies from 5 to 20.



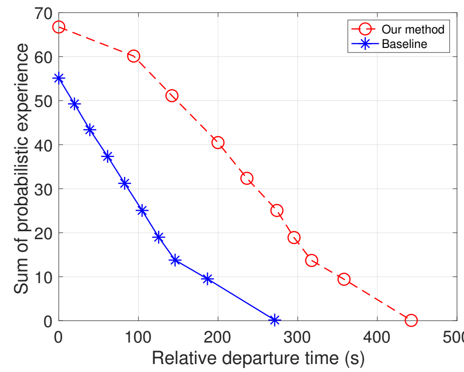
(c) Anti-correlated dataset: Comparison of workers' quality between our method and baseline based on $S_{\mathcal{E}}$ with window size N_{win} varies from 5 to 20.

Figure 5.3: Comparison and evaluation of workers' quality between our method and baseline method for different datasets with varied window size N_{win} from 5, 10, 15 to 20.

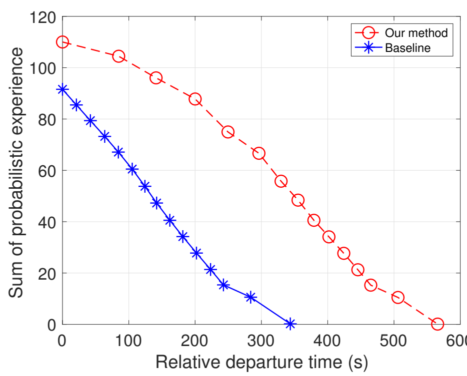
i) for each sliding window, the running time for worker selection is quite fast (from $0.5ms$ to $6.5ms$), which validates the efficiency and feasibility of our selection scheme. ii) The running time for worker selection is proportional to N_{win}, which equals to $\frac{(N_{win}+1)(N_{win}+2)}{2} \cdot t_{sky} + N_{win} \cdot t_{w_i,w_j}$. As a result, the more the number of workers in the current sliding window, the longer time it will take for selecting workers. iii) The computational cost is similar among different datasets. In the experiment, we simulate worker's $\mathcal{E}_i$ and T_i in each dataset. Essentially, the computational costs for securely comparing working experience in phase `CmpExp` and probabilistic skyline values in phase `CmpSky` account for a substantial part of the overall time costs in worker selection, which leads to the result that all the datasets have a similar



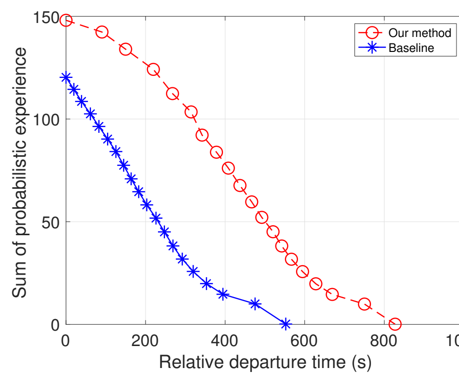
(a) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathtt{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 5$.



(b) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathtt{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 10$.

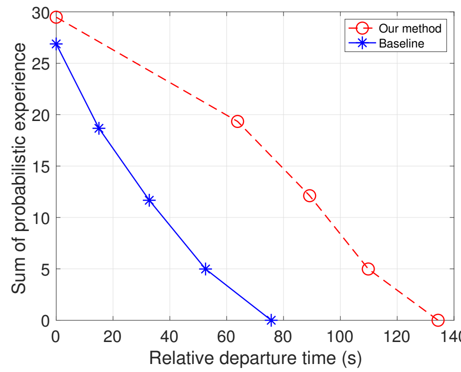


(c) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathtt{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 15$.

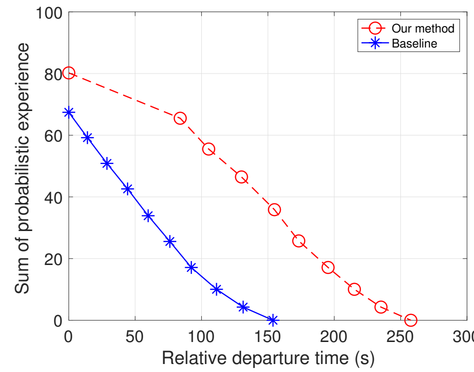


(d) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathtt{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 20$.

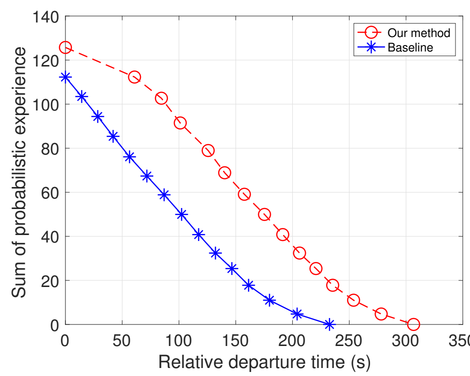
Figure 5.4: Real dataset: The sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathtt{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with varied window size $N_{win} = 5$, 10, 15 and 20, i.e., for a given window size N_{win}, $S_{\mathcal{E}}^{\mathtt{r}} = S_{\mathcal{E}} - \mathcal{E}_i$ is measured after a worker w_i leaves the system at $t_i^{\mathtt{rexp}}$.



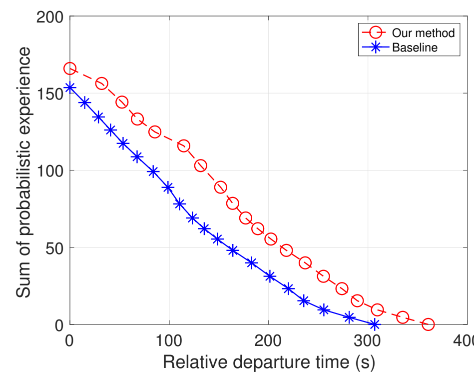
(a) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathfrak{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 5$.



(b) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathfrak{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 10$.



(c) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathfrak{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 15$.



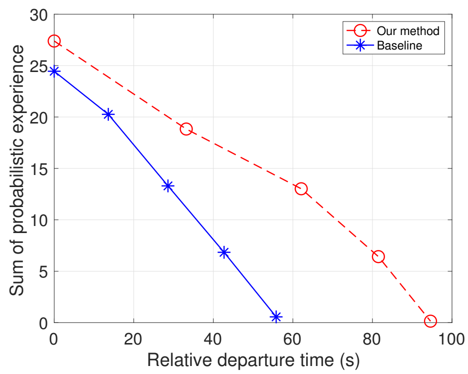
(d) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathfrak{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 20$.

Figure 5.5: Independent dataset: The sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathfrak{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with varied window size $N_{win} = 5$, 10, 15 and 20, i.e., for a given window size N_{win}, $S_{\mathcal{E}}^{\mathfrak{r}} = S_{\mathcal{E}} - \mathcal{E}_i$ is measured after a worker w_i leaves the system at $t_i^{\mathtt{rexp}}$.

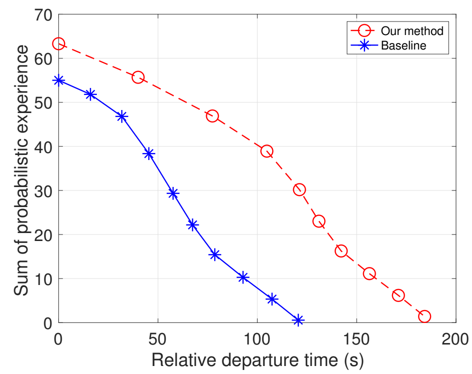
running time when N_{win} is confirmed.

• Two scenarios for performance evaluation: For both the baseline and our proposed scheme, the following two real-world scenarios are deployed on all the datasets, i.e., *scenario 1: workers continuously arrive at the system* and *scenario 2: workers continuously leave the system.*

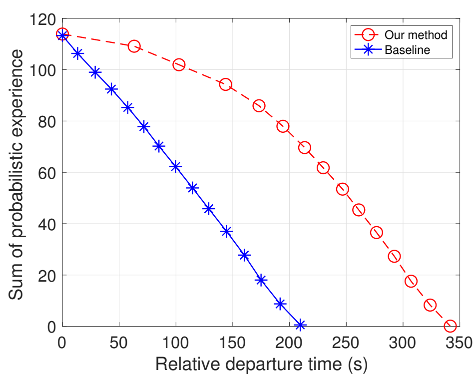
In *scenario 1*, with the successive arrival of the workers, the proposed worker selection scheme is maintained constantly to choose the top-N_{win} workers among the $N_{win}+1$ candidates. The sum of probabilistic worker experience $S_{\mathcal{E}} = \sum_{i=1}^{N_{win}}(T_i \times \mathcal{E}_i)$ is



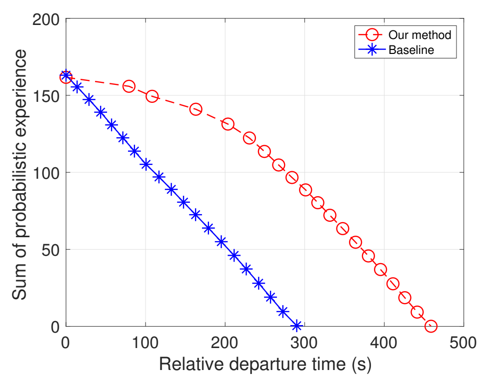
(a) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathtt{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 5$.



(b) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathtt{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 10$.



(c) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathtt{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 15$.



(d) Sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathtt{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with $N_{win} = 20$.

Figure 5.6: Anti-correlated dataset: The sum of probabilistic worker experience $S_{\mathcal{E}}^{\mathtt{r}}$ vs. relative departure time $t_i^{\mathtt{rexp}}$ with varied window size $N_{win} = 5$, 10, 15 and 20, i.e., for a given window size N_{win}, $S_{\mathcal{E}}^{\mathtt{r}} = S_{\mathcal{E}} - \mathcal{E}_i$ is measured after a worker w_i leaves the system at $t_i^{\mathtt{rexp}}$.

used as the evaluation metric in this scenario. $S_{\mathcal{E}}$ can be used to represent workers' overall quality in each sliding window, so it is a good indicator for the reliability of the proposed scheme. $S_{\mathcal{E}}$ is calculated for the selected N_{win} workers, and will be updated after a new worker arrives.

In *scenario 2*, for each current window, we assume that no worker arrives at the system anymore, and each worker leaves the MCS platform after they reach their expiry time. More specifically, for each worker w_i in the system, the following two variables are calculated: i) $t_i^{\mathtt{rexp}} = t_i^{\mathtt{exp}} - t_{min}^{\mathtt{exp}}$ which means w_i's relative departure time, where $t_{min}^{\mathtt{exp}}$ is the earliest expiry time in current window, and ii) $S_{\mathcal{E}}^{\mathtt{r}} = S_{\mathcal{E}} - \mathcal{E}_i$ which means the sum of probabilistic worker experience after w_i leaves the system. Consequently, *scenario 2* provides a useful circumstance for us to identify the overall sustainability and reliability of the system in terms of working experience and persevering duration when no worker arrives.

• Experimental results for *scenario 1*: Fig. 5.3 shows the experimental results for *scenario 1*. First of all, it is obvious that our proposed scheme performs better than the baseline method on selecting workers with more working experience. Specifically, $S_{\mathcal{E}}$ of our approach is larger than the baseline method for almost all the cases. This result indicates that given the same N_{win}, our scheme can keep each sliding window more reliable by maintaining workers with more experience. There even exists some cases such that the $S_{\mathcal{E}}$ of our proposed method when $N_{win} = 15$ is larger than baseline method when $N_{win} = 20$ (e.g., Fig. 5.3 (a) and (b)). Second, we can see that there is an exceptional example in Fig. 5.3 (c), where two methods are nearly indistinguishable for anti-correlated dataset when $N_{win} = 5$. Under the anti-correlated situation, a worker with more working experience tends to have a shorter life-span of the sensing data, and vice versa. This property leads to a result that when N_{win} is small (e.g. equals 5), the workers are more likely not to be dominated by each other. Thus, their $\Pr^{\mathtt{sky}}(w_i)$ tends to not be affected by the newly arrived worker (e.g., the case 2 in the `WrkSel` phase). So the workers may just be selected only by their expiry time. As a result, our proposed scheme shows a similar statistical pattern with the baseline. Third, as N_{win} increases, the differences of $S_{\mathcal{E}}$ between our scheme and baseline increase simultaneously. Consequently, it is important to define a sufficiently large N_{win} for better distinguishing our scheme with baseline in real-world MCS applications.

• Experimental results for *scenario 2*: Fig. 5.4 to Fig. 5.6 show the comparison results of *scenario 2* with N_{win} varies from 5 to 20. Intuitively, our proposed worker selection scheme is superior to the baseline method for all the cases in terms of $t_i^{\mathtt{rexp}}$ and $S_{\mathcal{E}}^{\mathtt{r}}$. This indicates that given the same N_{win}, our approach can provide better and longer services than baseline. Specifically, once a worker reaches the expiry time and leaves the platform, the rest of the workers in our system have more working experience than those in baseline method. In addition, the workers in our proposed selection scheme can stay 30s $\sim$ 200s longer than baseline workers under different N_{win}. Therefore, when no worker arrives anymore, the workers selected by our scheme can maintain the platform better and longer, which is significant for certain MCS services such as traffic jam/accident monitoring. This observation further validates our proposed scheme's reliability and sustainability in a special case, where the platform's durability plays a fundamental role in service quality.

5.6 Related Work

Recently, there have been numerous endeavors that aim to study the problem of worker selection in MCS platforms. In this section, we briefly review some of the typical works in this area.

Jin et al. [45] proposed a privacy-preserving framework for MCS worker selection based on a novel incentive mechanism. Specifically, their framework can compensate workers' costs for both sensing leakage and privacy leakage while selecting workers who are more likely to provide reliable sensing data. Moreover, they integrated their framework with data perturbation and aggregation technologies, ensuring highly accurate aggregated results and guarantees workers' privacy. Zhang et al. [136] proposed a probabilistic skyline based worker selection method, which solved a fundamental problem of calculating workers' trustability based on their historical reviews. In addition, they designed a non-interactive encrypted integer comparison protocol to compare the skyline dominance relationship between workers securely. Skyline operator is used for selecting workers based on ask price and task relativity. This is the first work using a skyline-based method for worker selection in MCS platforms, which can consider the trade-off between multiple aspects and select workers that are not dominated by others. Gong et al. [32] devised truthful crowdsensing mechanisms for incentivizing strategic workers to truthfully reveal their private quality and truthfully make efforts as desired by the requester. In their work, the authors assumed that a strategic worker with low quality may pretend to have a high quality to receive a high reward from the requester. Under the proposed mechanisms, they showed that the requester can assign the task only to the best workers that has the smallest virtual valuation. However, in our work, by introducing a performance feedback mechanism for calculating trustability, the workers need to submit the real-time information and sensing data as accurate as possible. Ren et al. [90] introduced a socially aware reputation management scheme for selecting well-suited participants and allocating the task rewards in MCS services. Concretely, social attributes, task delays, and reputation are focused under a fixed task budget. Moreover, a rewarding scheme is devised to measure the quality of the sensing reports and allocate reliable workers to certain MCS tasks under the consideration of the trustworthiness and cost performance of task participants. Ni et al. [75] described a privacy-preserving MCS framework for location-based applications, which can balance the trade-off between privacy preservation and task allocation. A matrix-based location matching mechanism is used by the service provider to achieve location-based task allocation without disclosing workers' sensing location. Proxy re-encryption technology is developed to enhance privacy preservation during the MCS services. Jin et al. [44] proposed a differentially private incentive mechanism that preserves the privacy of each worker's bid, which is based on the single-minded reverse combinatorial auction. However, in this work, we select workers based on working experience, expiry time, and trustability, without considering workers' asking price information. Liu et al. [63] studied the problem of multi-task allocation in MCS services. For particular, two typical multi-task MCS environments are considered in their work, e.g., FPMT (few participants with more tasks) and MPFT (more participants with few tasks). Unique mechanisms are devised for different scenarios with different optimization goals. Wang et al. [107] proposed a location privacy-preserving worker selection framework with geo-obfuscation for protecting workers' locations during task allocation. More specifically, by obfuscating their real locations under the guarantee of differential privacy, workers can protect their location privacy regardless of adversaries' prior knowledge. In addition, a mixed-integer non-linear programming problem is defined for minimizing workers' expected travel distance and optimizing worker assignments. Wu et al. [116] proposed a context-aware multi-armed bandit incentive method for selecting high-quality workers in MCS systems. In their work, workers' service quality is evaluated by their context and cost. Then, by accurately assessing workers' quality information, a modified Thompson sampling approach is utilized for selecting reliable workers based on reinforcement learning. Sun et al. [100] studied the problem of truth discovery in crowdsourced question answering system based on a contract-based privacy-preserving incentive mechanism, whereas the scope of this work is reliable and continuous worker selection in MCS.

In summary, most of the previous works treat the MCS task as a one-time service.

They allocate tasks to suitable workers by measuring their personal qualification, location, and trustability in particular sensing environment (e.g., multi-task allocation), without considering a dynamic situation such that workers may continuously arrive or leave the system. Furthermore, little systematic and data-driven evidence has been published for a typical real-world case in which the MCS task should be constantly maintained for a long time. Unlike the above, our work focuses on privacy-preserving and continuous worker selection for MCS platforms. Our proposed scheme can dynamically select reliable workers while guarantee privacy preservation of their sensitive information.

5.7 Concluding Remarks

In this chapter, we have proposed a privacy-preserving worker selection scheme for MCS based on probabilistic skyline query over sliding windows. The proposed scheme can continuously select reliable workers in terms of working experience, expiry time, and trustability without revealing sensitive information. More specifically, a probabilistic skyline approach is designed for a dynamic situation where workers may constantly arrive at/leave the platform. For protecting workers' privacy, we have designed an ElGamal-based encryption approach for securely outsourcing sensitive information and comparing workers' personal information. Security analysis demonstrates that the proposed scheme is privacy-preserving. Extensive experiments have been conducted on both real-world and simulated datasets for two scenarios: i) continuous worker arrival and ii) continuous worker departure. The comparison results validate the efficiency and effectiveness of our proposed worker selection scheme. For future work, the group skyline technique will be studied to enhance the performance of the current system.

Chapter 6

FedSky: An Efficient and Privacy-preserving Scheme for Federated Mobile Crowdsensing

6.1 Introduction

In the previous two chapters, we proposed privacy-preserving schemes in terms of both static and dynamic worker selection processes. The protected information is only used in worker selection, such as workers' reviews, working experience, expiry time, and trustability scores. However, workers' sensing data may also contain sensitive information and should be protected before being uploaded to the MCS platform. For example, a 3D landmark reconstruction task requires each worker to take 2D photos of a landmark and submit the photos to the MCS platform [16]. In such case, the uploaded photos may contain workers' sensitive information, e.g. driver license and real-time location information. Therefore, the question now is how to protect workers' sensing data from being disclosed to untrusted parties.

Up until now, a large and growing body of literature has been proposed for solving these privacy challenges in MCS. Among all the studies, Federated Learning (FL) can be considered a potential and practical solution. In FL, the platform iteratively selects random workers to download a trainable model [56]. Then, the selected workers update the model with their own data, and upload the updated model to the platform. After that, the platform aggregates multiple updates to further improve the model [77]. The distributed nature of FL enables the workers to optimize the shared model while keeping all the training data locally, thus ensuring their privacy. Based on [46, 127], FL can be broadly categorized into *cross-silo* FL and *cross-device* FL. In *cross-silo* FL, the workers are functional organizations (e.g., financial agencies) with abundant computing resources. The trained model is exclusively released to these organizations, but not the FL aggregation platform. In contrast, in the *cross-device* FL, the workers are heterogeneous mobile users with limited computational powers. The FL platform will finally get access to the trained model. In this chapter, by introducing *cross-device* FL into MCS, we propose a novel sensing scenario called Federated MCS (F-MCS). F-MCS allows workers to build a robust and secure machine learning model without sharing their sensing data. As a result, F-MCS can address the critical issue of workers' information leakage in traditional MCS systems and is expected to become a new hotspot of mobile sensing services.

Nevertheless, there are still two major issues if we simply integrate the FL technique with MCS services. (1) The selection of stable workers in F-MCS is challenging. In F-MCS, the mobile workers are heterogeneous with different computational capacities and data resources. For example, their devices are different in terms of hardware (CPU, memory), network connectivity (3G, 4G, 5G, Wi-Fi, signal strength), and power status (battery level, charging capacities). The synchronous training protocol [127] is widely used in FL, where no worker can proceed to the next training round until all workers' data have been uploaded. The workers with limited computational powers require a longer time to update the model parameters. Therefore, such a problem will delay the subsequent aggregation step and make the overall training process inefficient. (2) The lack of practical privacy-preserving solutions for *cross-device* F-MCS platforms. Even though the workers can avoid leaking their sensing data in F-MCS, they still face possible information disclosure. Specifically, the shared model information (e.g., the gradient) uploaded by the workers can leak important statistical patterns of the local datasets [39, 3, 69]. Consequently, homomorphic encryption (HE)-based privacy-preserving data aggregation techniques have been well-studied for enhancing the privacy of FL platforms [124]. However, most of the approaches are designed for *cross-silo* FL setting where the final global model can only be accessed by the participants (i.e., workers) [54, 67, 122, 25, 39, 127, 93]. In contrast, F-MCS is a *cross-device* FL system where the final model will be released to the F-MCS platform for certain real-time services. Hence, the existing methods cannot be applied in F-MCS. There is an immediate necessity for designing practical and privacy-preserving solutions for F-MCS applications.

In this chapter, aiming to solve the above challenges, we propose a novel and secure data aggregation scheme for F-MCS applications, called FedSky. By extending the classic FedAvg algorithm [70], FedSky is characterized by employing group skyline for selecting qualified workers and homomorphic encryption (HE) for securely aggregating data. Specifically, the main contributions of this chapter are threefold as follows.

- First, we propose a novel and effective worker selection mechanism in FedSky for F-MCS. Concretely, in each communication round, instead of choosing a group of workers at random, we select a skyline group of workers in terms of workers' local data sizes and their mobile devices' computational powers. In this way, compared with the traditional FedAvg algorithm, our approach can significantly reduce workers' computational time and the platform's waiting time, and therefore significantly maximize the efficiency of the FL process.

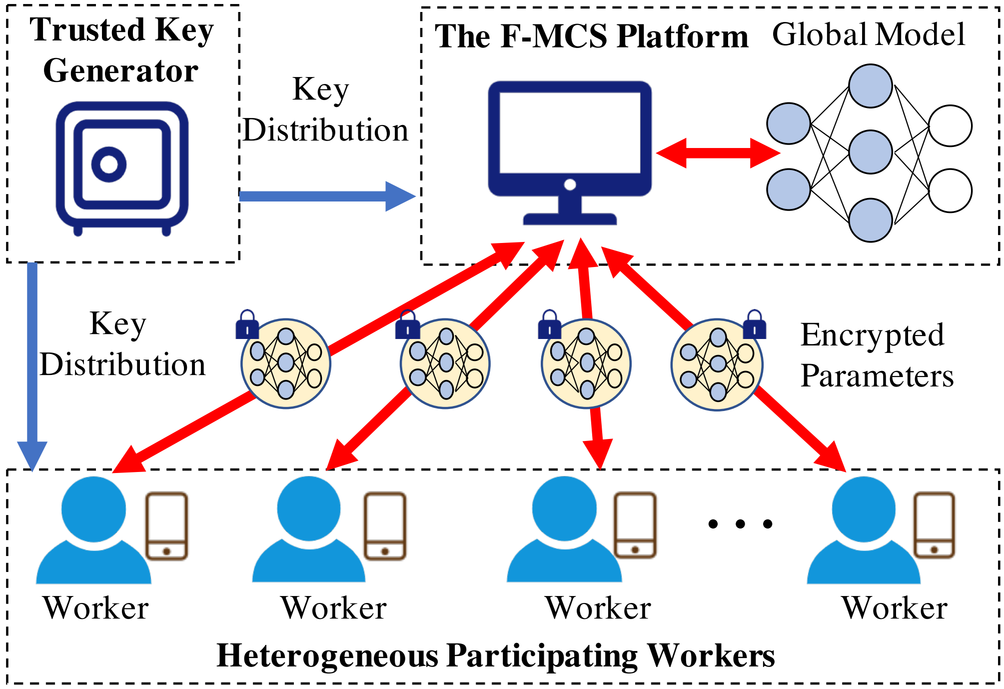


Figure 6.1: The system model under consideration in Chapter 6.

- Second, we propose a novel privacy-preserving data aggregation scheme for F-MCS platform based on the HE technique. The scheme is designed for *cross-device* FL setting such that the final global model can be accessed by the F-MCS platform. In addition, the overall training process requires no interaction between the selected workers. Except for the left neighbor ID and right neighbor ID, each selected worker knows nothing about other workers' information.

- Third, we build a custom simulator for performance evaluation. Extensive experimental results validate the effectiveness and efficiency of the proposed schemes. In particular, by introducing a novel worker selection mechanism and a privacy-preserving data aggregation protocol, we can greatly improve the efficiency of the FL process without affecting its accuracy.

The remainder of this chapter is organized as follows. In Section 6.2, we introduce our system model, security model and design goals. In Section 6.3, we propose our scheme in details. Then in Section 6.4 and Section 6.5, we present the security analysis and performance evaluation, followed by the related work in Section 6.6. Finally, we recap the conclusions in Section 6.7.

6.2 Models and Designed Goals

In this section, we first formalize our system model and security model, and then identify our design goals. For reference, a summary of frequently-used notations is given in Table 6.1.

6.2.1 System Model

In our system model (see in Fig. 6.1), we mainly consider a typical F-MCS scenario including three entities, namely, a trusted key generator $\mathcal{TKG}$, an F-MCS platform $\mathcal{P}$, and a group of heterogeneous workers $\mathcal{W} = \{w_1, w_2, ...\}$.

• The Trusted Key Generator ($\mathcal{TKG}$): $\mathcal{TKG}$ is a trusted authority that generates and distributes proper keys to the corresponding entities so that a certain F-MCS task can be completed cooperatively.

• The F-MCS Platform ($\mathcal{P}$): $\mathcal{P}$ is the trusted platform for providing F-MCS services and is responsible for performing an F-MCS task, including registering workers, initializing the task model, selecting workers, and training the model. More specifically, upon registration, $\mathcal{P}$ assigns an unique identifier $\texttt{ID}_{w_i}$ to each registered worker w_i, and then broadcasts the list of workers' identifiers to $\mathcal{TKG}$. In the training process, $\mathcal{P}$ first initializes the global model and selects a fraction of qualified workers from $\mathcal{W}$ (details of worker selection will be introduced in Section 6.3.3). Next, $\mathcal{P}$ distributes the model to the selected workers and trains the model by aggregating the worker-provided model parameters in each training round. Multiple rounds of interactions between $\mathcal{P}$ and the selected workers are demanded to achieve the final global model.

- The Participating Workers $\mathcal{W} = \{w_1, w_2, \cdots\}$: $\mathcal{W}$ are the participating workers who wish to conduct a certain F-MCS task. If a worker in $\mathcal{W}$ is selected by $\mathcal{P}$, at first, he/she needs to collect the raw sensing data using his/her mobile device(s). After receiving the global model, each selected worker needs to update the model parameters by training the model with his/her local data, encrypt the model parameters, and exchange the ciphertexts with $\mathcal{P}$. To be selected in each training round, the workers need to periodically send their current training capacities and computational statuses to $\mathcal{P}$, e.g., the size of the local dataset, the current CPU share/battery/memory of their mobile devices.

6.2.2 Security Model

First, we consider $\mathcal{P}$ and $\mathcal{TKG}$ are fully trusted. Notably, some malware have been installed in $\mathcal{P}$ by an adversary $\mathcal{A}$ without being detected. Therefore, $\mathcal{A}$ can monitor $\mathcal{P}$'s database and eavesdrop on the communication information between $\mathcal{P}$ and $\mathcal{W}$. Essentially, $\mathcal{A}$ is interested in the model updates that each worker uploads to the platform. With those updates, $\mathcal{A}$ may infer workers' real-time spatial-temporal information. Besides, we consider the workers $\mathcal{W}$ are honest-but-curious. So all the workers strictly follow the designed protocols for updating, encrypting, and uploading model parameters but may be interested in other workers' data resources. Moreover, we assume that there is no collusion among workers in $\mathcal{W}$, and workers cannot collude with $\mathcal{P}$ either. It is worth noting that outside attackers may exploit other vulnerabilities of the F-MCS platform. As this work focuses on privacy-preservation, those attack are beyond this thesis's scope and will be discussed in our future work.

6.2.3 Design Goals

Based on the system model and security model mentioned above, our design goal is to develop an efficient and privacy-preserving scheme for F-MCS applications.

Table 6.1: The summary of important notations used in chapter 6

Notation	Definition
System Notations	
$\mathcal{TKG}$	The trusted key generator
$\mathcal{P}$	The F-MCS platform
$\mathcal{W} = \{w_1, w_2, ..., w_K\}$	The participating workers
Skyline Notations	
G-skyline	Group skyline
C-skyline	Constrained skyline
$\mathcal{C} = \{\texttt{Con}_1, \ldots ..., \texttt{Con}_d\}$	A set of constrains
$w_{ui} \preceq w'_{vi}$	$w_{ui} \prec w'_{vi}$ or $w_{ui} = w'_{vi}$
CG-skyline	Constrained group skyline
FL Notations	
$\mathcal{D}_i$	Worker w_i's local data
x_i	Worker w_i's model parameter
y_i	Weight for x_i
g_i	Worker w_i's local gradient for x_i
Model Notations	
N_i	Worker w_i's data size, i.e., $N_i = \|\mathcal{D}_i\|$
P_i	Worker w_i's computational power
$\mathcal{W}_s = \{w_1, w_2, ..., w_k\}$	The group of selected k workers

Specifically, the following desirable properties should be achieved.

• *Efficiency*: The proposed FedSky scheme should be efficient in terms of training models and uploading the encrypted model parameters in each communication round. Consequently, compared with the traditional FedAvg algorithm, $\mathcal{P}$'s waiting time should be shortened and the overall FL training efficiency should be improved in our scheme.

• *Privacy-preservation*: We plan to design a privacy-preserving F-MCS framework, which can prevent the confidentiality of workers' sensitive information from being disclosed. More specifically, for $\forall w_i$, let $\mathcal{D}_i$ denote w_i's local data and x_i denote w_i's updated model parameters after each training round, both $\mathcal{D}_i$ and x_i should be kept secret to other workers. Moreover, even if $\mathcal{A}$ eavesdrops $\mathcal{P}$'s database and steals the communication data between $\mathcal{W}$ and $\mathcal{P}$, it still cannot identify each worker's uploaded model parameters x_i.

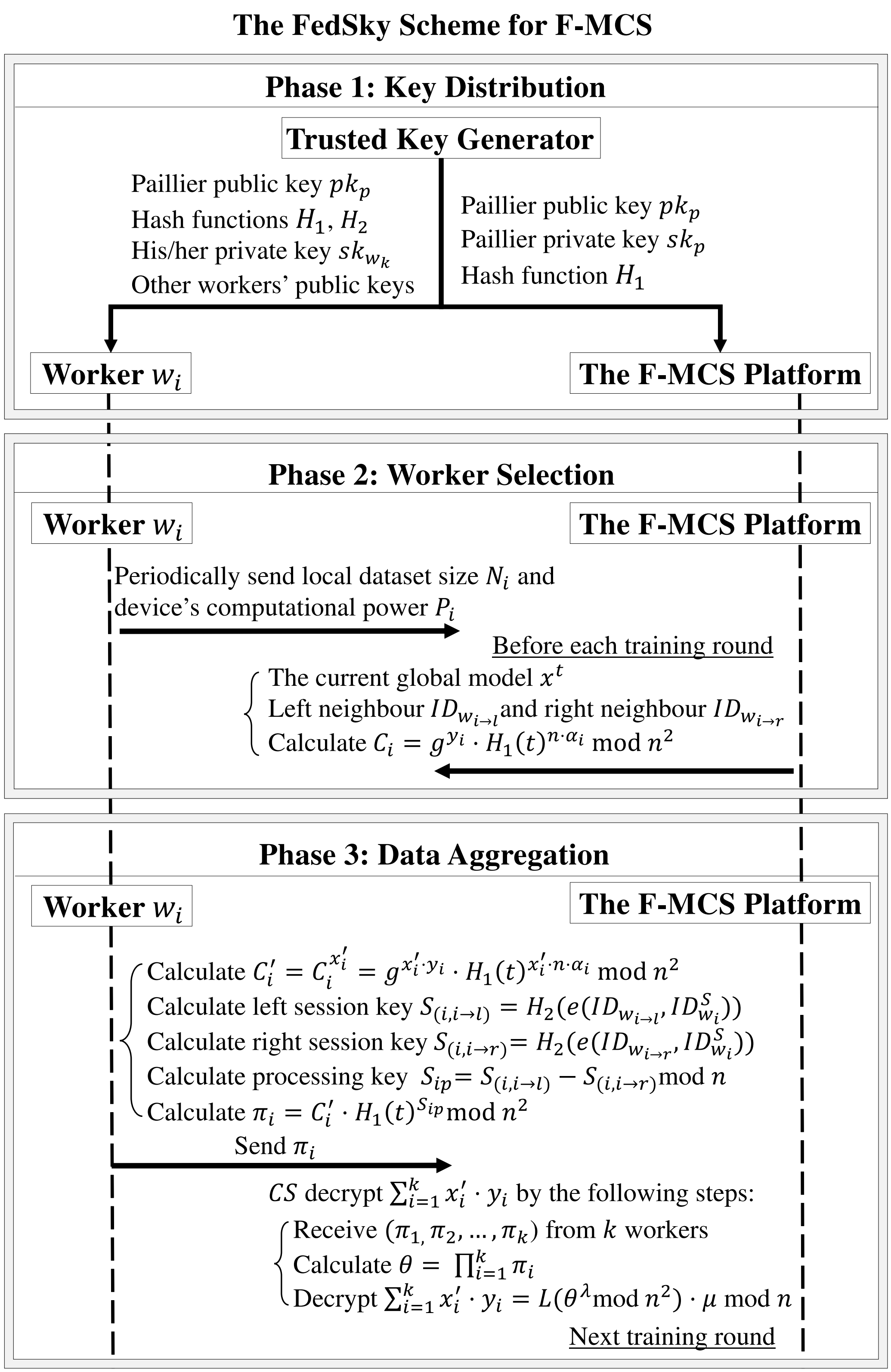


Figure 6.2: The overall workflow of the FedSky scheme.

6.3 The Proposed Scheme

In this section, we propose a novel data aggregation scheme for F-MCS called FedSky based on the classic FedAvg algorithm. We first introduce the basic information about FedAvg. After that, we describe the proposed scheme in details.

Algorithm 9 The FedAvg algorithm [124]

Data aggregator executes: (Executed by the data aggregator)
for each training round $t = 1, 2, ...$ **do**
  Randomly selects k workers from all the K candidates
  for each selected worker w_i **in parallel do**
    $x_i \leftarrow$ **WorkerUpdate**(i, x^t)
  end for
  Takes the weighted average of the received model parameters from the k workers: $x^{t+1} \leftarrow \sum_{i=1}^{k} \frac{N_i}{N} x_i$
  if The model parameters converge **then**
    Training finishes
  end if
  if The model parameters do not converge **then**
    Broadcasts the aggregated model parameters x^{t+1} to all the workers
  end if
end for
WorkerUpdate(i, x^t): (Executed by each selected worker)
for each local epoch e from 1 to E **do**
  batches $\leftarrow$ randomly divides dataset $\mathcal{D}_k$ into batches of size S
  for each batch b in batches **do**
    Computes the batch gradient g_i^b
    Updates model parameter by $x_i = x^t - \eta \cdot g_i^b$
  end for
end for
Send the local model parameters x_i to the data aggregator

6.3.1 The FedAvg Algorithm

The FedAvg [70] is a federated optimization algorithms for training ML/DL models in FL. Essentially, there are two entities in a typical FL setting, they are (1) data aggregator and (2) K participating workers $\{w_1, w_2, ..., w_K\}$. Let N_i denote the number of datasets that owned by worker w_i. In FedAvg, at the beginning of each round t, the data aggregator randomly selects k workers from all the K candidates, and then broadcasts the current global model x^t (at round t) to each of the k workers. After receiving the current model, worker w_i computes the average gradient on its local dataset, which is denoted by g_i. Next, given a fixed learning rate η, w_i calculates $x_i \leftarrow x^t - \eta \cdot g_i$ and uploads x_i back to the data aggregator. Finally, the data aggregator updates the global model by $x^{t+1} \leftarrow \sum_{i=1}^{k} \frac{N_i}{N} x_i$, where $N = N_1 + N_2 + \ldots N_k$. The details of FedAvg is shown in Algorithm 9.

Our proposed scheme mainly consists of three phases: Key Distribution ($\mathsf{KeyDist}$), Worker Selection ($\mathsf{WorkSel}$), and Data Aggregation ($\mathsf{DataAgg}$). In $\mathsf{KeyDist}$ phase, $\mathcal{TKG}$ will generate keys and then distribute them to the correct entities. In $\mathsf{WorkSel}$ phase, $\mathcal{P}$ will select an optimal group of workers by k-point CG-skyline. Finally, in $\mathsf{DataAgg}$ phase, the selected workers need to train the model, encrypt the updated model parameter x_i and securely send it to $\mathcal{P}$. At last, $\mathcal{P}$ decrypts the ciphertexts to get the final aggregation results. The overall workflow of FedSky is shown in Fig. 6.2. The detailed description of each phase is given as follows.

6.3.2 The $\mathsf{KeyDist}$ Phase

First, $\mathcal{TKG}$ computes (p, q, n, λ, μ) by running $\texttt{KeyGen}(\kappa)$, and generates the public key $pk_p = (n, g)$, and the private key $sk_p = (\lambda, \mu)$. Next, $\mathcal{TKG}$ chooses an uniform cryptographic hash function H_1 where $H_1 : \{0,1\}^* \rightarrow \mathbb{Z}_{n^2}$ and a bilinear pairing map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. After that, $\mathcal{TKG}$ broadcasts $(pk_p, H_1, e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T)$ to $\mathcal{P}$ and all the workers in the system, and securely sends sk_p to $\mathcal{P}$.

In the next step, $\mathcal{TKG}$ chooses another cryptographic hash function H_2 such that $H_2 : \{0,1\}^* \rightarrow \mathbb{Z}_n$. Then, $\mathcal{TKG}$ chooses a random number $S \in \mathbb{Z}_n^*$ as the master key. For $\forall w_i \in \mathcal{W}$, given w_i's identifier $\texttt{ID}_{w_i}$ where $\texttt{ID}_{w_i} = H_0(w_i)$ and $H_0 : \{0,1\}^* \rightarrow \mathbb{G}$, $\mathcal{TKG}$ calculates $\texttt{ID}_{w_i}^S$. Last, $\mathcal{TKG}$ shares H_2 to all the workers and securely sends $\texttt{ID}_{w_i}^S$ to w_i.

Algorithm 10 CG-skyline based worker selection

Input: A worker database $\mathcal{W} = \{w_1, w_2, ..., w_K\}$ where $w_i = (N_i, P_i)$ for $i \in [1, K]$, a given set of two constraints $\mathcal{C} = \{\texttt{Con}_1, \texttt{Con}_2\}$
Output: A set of selected workers with size k
$\mathcal{P}$ finds the set of workers $\mathcal{W}'$ based on $\mathcal{C} = \{\texttt{Con}_1, \texttt{Con}_2\}$
$\mathcal{P}$ initializes an empty Max Priority Queue $\texttt{Q}_{\mathcal{W}}$
for $\forall w_i \in \mathcal{W}'$ **do**
    $\mathcal{P}$ calculates $\texttt{Sum}(w_i) = N_i + P_i$
    $\mathcal{P}$ adds w_i into $\texttt{Q}_{\mathcal{W}}$ by the value of $\texttt{Sum}(w_i)$
end for
$\mathcal{P}$ initializes another empty Max Priority Queue $\texttt{Q}_{non}$ and an empty list $\texttt{Q}_{sky}$
$\mathcal{P}$ removes the first worker in $\texttt{Q}_{\mathcal{W}}$ and adds this worker into $\texttt{Q}_{sky}$
for $\forall w_i \in \texttt{Q}_{\mathcal{W}}$ **do**
    $\mathcal{P}$ removes w_i from $\texttt{Q}_{\mathcal{W}}$
    $\mathcal{P}$ compares the dominance relationship between w_i and all the workers in $\texttt{Q}_{sky}$
    if $\exists\ w_j \in \texttt{Q}_{sky}$ s.t. $w_j \prec w_i$ **then**
        $\mathcal{P}$ adds w_i into $\texttt{Q}_{non}$
    end if
    if No such w_j exists **then**
        $\mathcal{P}$ adds w_i into $\texttt{Q}_{sky}$
    end if
end for
if $k' < k$ where k' is number of workers in $\texttt{Q}_{sky}$ **then**
    $\mathcal{P}$ adds the first $k - k'$ workers from $\texttt{Q}_{non}$ into $\texttt{Q}_{sky}$
end if
return The workers in $\texttt{Q}_{sky}$

6.3.3 The WorkSel Phase

In this chapter, each worker w_i has two attributes: (1) the size of w_i's local dataset N_i, and (2) w_i's mobile device's computational power P_i. Same as [77], we assumed that P_i is measured by how many data samples a worker can process per minute for the F-MCS task. Each worker w_i is required to periodically send N_i and P_i to the platform $\mathcal{P}$. Before worker selection, $\mathcal{P}$ first defines a set of 2-dimensional constraints $\mathcal{C} = \{\texttt{Con}_1, \texttt{Con}_2\}$ based on the F-MCS task requirements. For higher requirements (e.g., higher accuracy and less training latency), the constraints $\texttt{Con}_1$ and $\texttt{Con}_2$ should be chosen more strictly. Specifically, for $\texttt{Con}_1 = [min_1, max_1]$ and $\texttt{Con}_2 = [min_2, max_2]$, we say $\texttt{Con}_1$ is more strictly than $\texttt{Con}_2$ if $min_1 > min_2$ and $max_1 > max_2$. After that, $\mathcal{P}$ selects the set of workers $\mathcal{W}'$ for $\mathcal{W}' \subseteq \mathcal{W}$ whose current information matches the constraints.

Before each training round, $\mathcal{P}$ will select an optimal group of k workers from $\mathcal{W}'$ by CG-skyline in terms of N_i and P_i. The details of the proposed approach are discussed as follows.

For each worker w_i in $\mathcal{W}'$, $\mathcal{P}$ first calculates $\texttt{Sum}(w_i)$ where $\texttt{Sum}(w_i) = N_i + P_i$. Then, all the workers are added in a pre-initialized Max Priority Queue $\texttt{Q}_{\mathcal{W}}$ by $\texttt{Sum}(w_i)$, where all the workers are sorted by $\texttt{Sum}(w_i)$ in a descending order and the workers with larger $\texttt{Sum}(w_i)$ values will be removed earlier from $\texttt{Q}_{\mathcal{W}}$ than the workers with smaller $\texttt{Sum}(w_i)$ values. It is worth noting that the magnitudes of P_i and N_i should be in the same level, such that the sum of P_i and N_i is not dominated by any of the two attributes. Based on the definition of skyline discussed in Section 3.1.4, we have the following theorems.

Theorem 6.3.1. *Let w_i denote a worker with N_i is the number of data samples owned by w_i and P_i is w_i's computational power. If w_i is the first worker being removed from $\texttt{Q}_{\mathcal{W}}$, then w_i is a skyline worker.*

Proof. Assume there exists another worker w_i' such that $w_i' \prec w_i$. Let N_i' and P_i'

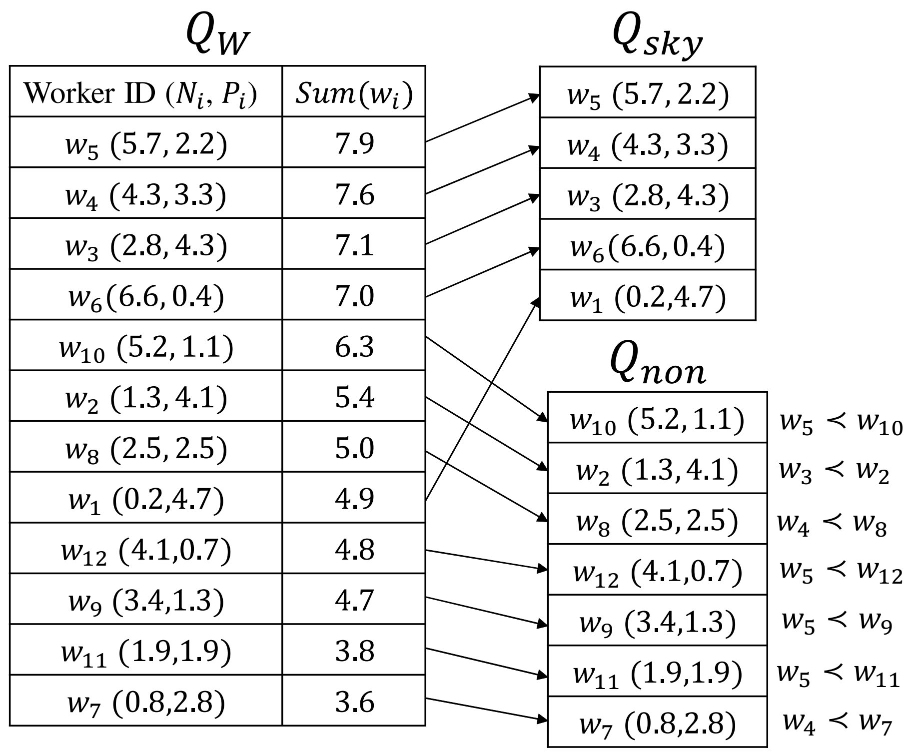


Figure 6.3: An illustration example of the proposed worker selection approach for CG-skyline computation.

denote the number of data samples owned by w_i' and w_i''s computational power, respectively. Based on the definition of skyline, w_i' is better than w_i for at least one attribute, and not worse than w_i for the rest of attributes. It is easy to know that $N_i' + P_i' > N_i + P_i$, i.e., $\mathtt{Sum}(w_i') > \mathtt{Sum}(w_i)$. Therefore, w_i' should be removed first from $\mathtt{Q}_{\mathcal{W}}$, which contradicts our assumption that w_i is the first worker to be removed. Therefore, such w_i' does not exist and our theorem holds. $\square$

Theorem 6.3.2. *For $\forall w_j \in \mathcal{W}'$, all the workers who dominates w_j will be removed earlier from $\mathtt{Q}_{\mathcal{W}}$ than w_j.*

Proof. Let w_j' denote one of the workers who dominate w_j. Again, based on the definition of skyline, it is easy to have $\mathtt{Sum}(w_j') > \mathtt{Sum}(w_j)$. Consequently, w_j' will be removed earlier than w_j from $\mathtt{Q}_{\mathcal{W}}$. $\square$

Next, $\mathcal{P}$ initializes another empty Max Priority Queue $\mathtt{Q}_{non}$ and an empty list $\mathtt{Q}_{sky}$.

Then, $\mathcal{P}$ goes through the workers in $\mathtt{Q}_{\mathcal{W}}$. By Property 1, the first worker will be removed from $\mathtt{Q}_{\mathcal{W}}$ and directly added into $\mathtt{Q}_{sky}$. The next removed worker from $\mathtt{Q}_{\mathcal{W}}$ will be compared with the workers(s) in $\mathtt{Q}_{sky}$. If he/she is not dominated by all the worker(s) in $\mathtt{Q}_{sky}$, then this worker will be added in $\mathtt{Q}_{sky}$. Otherwise, based on Property 2, $\mathcal{P}$ can safely add this worker in $\mathtt{Q}_{non}$. $\mathcal{P}$ repeats the steps above for the remaining workers until either the number of workers in $\mathtt{Q}_{sky}$ equals k or $\mathtt{Q}_{\mathcal{W}}$ is empty. Let k' denote the final number of workers in $\mathtt{Q}_{sky}$. If $k' < k$, then $\mathcal{P}$ removes the first $k - k'$ workers from $\mathtt{Q}_{non}$ and adds them into $\mathtt{Q}_{sky}$. In this way, we can select skyline workers as much as possible. The proposed worker selection approach is shown in Algorithm 10.

Example 6.3.1. *Fig. 3.3 shows a simple example of 12 workers. Each worker is represented by two attributes: (1) local data size and (2) local device's computational power. Based on Definition of skyline, we know that* $w_1, w_3, w_4, w_5,$ *and* w_6 *are the skyline workers. Fig. 6.3 shows a simple example of the proposed worker selection approach for CG-skyline query. The 12 workers are added in the Max Priority Queue* $\mathtt{Q}_{\mathcal{W}}$ *by* $\mathtt{Sum}(w_i)$. *After that, they are removed one by one from* $\mathtt{Q}_{\mathcal{W}}$ *and added into either* $\mathtt{Q}_{sky}$ *or* $\mathtt{Q}_{non}$. *In particular,* w_5, w_4, w_3, w_6, w_1 *are added into* $\mathtt{Q}_{sky}$ *because they are skyline workers. The rest of the workers (i.e.,* $w_{10}, w_2, w_8, w_{12}, w_9, w_{11}, w_7$*) are added into* $\mathtt{Q}_{non}$ *because they are dominated by at least one worker in* $\mathtt{Q}_{sky}$. *In this example,* $k' = 5$. *If* $k \leq k'$, *then the first* k *workers are the results for the k-point CG-skyline query. For example, if* $k = 3$, *then* $\{w_5, w_4, w_3\}$ *are selected. However, if* $k > k'$, *then all the workers in* $\mathtt{Q}_{sky}$ *plus the first* $k - k'$ *workers are the final results. For example, if* $k = 7$, *then* $\{w_5, w_4, w_3, w_6, w_1, w_{10}, w_2\}$ *are selected.*

It is worth noting that the proposed CG-skyline technique can only select one group skyline workers. It cannot find all the group skyline workers that are not dominated by other groups. After worker selection, the MCS platform $\mathcal{P}$ broadcasts the global model's hyperparameters (e.g., learning rate) to all the selected workers, which can

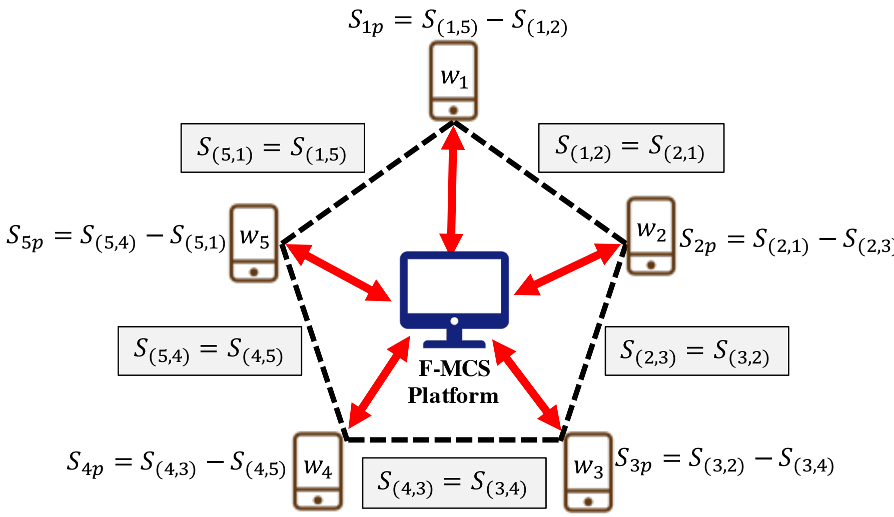


Figure 6.4: An example of 5 selected workers in a loop, where the workers have no interactions with each other. In the example, each worker w_i calculates S_{ip}, then we can simply get $\sum_{i=1}^{5} S_{ip} \bmod n = (S_{1p} + S_{2p} + S_{3p} + S_{4p} + S_{5p}) \bmod n = 0$.

be used by the workers to train and optimize the model in the next step.

6.3.4 The DataAgg Phase

Let $\mathcal{W}_s$ denote a group of k selected workers, i.e., $\mathcal{W}_s = \{w_1, w_2, ..., w_k\}$. At the beginning of each training round, for $\forall w_i \in \mathcal{W}_s$, $\mathcal{P}$ calculates $y_i = N_i/(N_1 + N_2 + ... + N_k)$ as the weight for w_i's model parameter. Each y_i is rounded to 3 decimal places. Next, $\mathcal{P}$ amplifies each y_i by 1000 to convert it as a positive integer, i.e., $y_i \leftarrow y_i \cdot 1000$. The detailed steps of the DataAgg phase are introduced as follows.

Step 1. $\mathcal{P}$ puts all the k workers from $\mathcal{W}_s$ in a loop. In the loop, w_{i+1} is w_i's right neighbor and w_{i-1} is w_i left neighbor for $i \in [2, k-1]$. Particularly, w_k is w_1's left neighbor; w_1 is w_k's right neighbor. Fig. 6.4 shows an example of such loop with 5 workers.

Step 2. $\mathcal{P}$ generates a random number $\alpha_i \in \mathbb{Z}_n^*$. Next, $\mathcal{P}$ sends the following information to w_i:

$$\begin{cases} \text{The current global model } x^t \text{(at round } t) \\ \text{the ID of } w_i\text{'s left neighbor } \mathtt{ID}_{w_{i\to l}}, \\ \text{the ID of } w_i\text{'s right neighbor } \mathtt{ID}_{w_{i\to r}}, \\ C_i = g^{y_i} \cdot H_1(t)^{n\cdot\alpha_i} \bmod n^2, \end{cases} \tag{6.1}$$

where $i \to l$ and $i \to r$ are indexes of w_i's left neighbor and right neighbor, respectively. Moreover, t is the current training round (e.g., t = 0, 1, 2, etc.). Therefore, the value of t changes in each training round.

Step 3. Upon receiving the above information, w_i first calculates the average gradient g_i based on his/her local dataset. Next, given a fixed learning rate η, w_i computes the updated local model parameter x_i by $x_i \leftarrow x^t - \eta \cdot g_i$. Besides, to keep the original information as much as possible, we assume that each worker w_i retains x_i to eight decimal places. So after each local training round, w_i needs to convert x_i to a positive integer $\bar{x}_i$ in $\mathbb{Z}_n$ by $\bar{x}_i = 10^8 \cdot x_i \bmod n$.

Step 4. Next, w_i computes $\bar{C}_i$, the left session key $S_{(i,i\to l)}$, the right session key $S_{(i,i\to r)}$, and the processing key S_{ip} as

$$\begin{cases} \bar{C}_i = C_i^{\bar{x}_i} = g^{\bar{x}_i\cdot y_i} \cdot H_1(t)^{\bar{x}_i\cdot n\cdot\alpha_i} \bmod n^2, \\ S_{(i,i\to l)} = H_2(e(\mathtt{ID}_{w_i}^S, \mathtt{ID}_{w_{i\to l}})) = H_2(e(\mathtt{ID}_{w_i}, \mathtt{ID}_{w_{i\to l}})^S), \\ S_{(i,i\to r)} = H_2(e(\mathtt{ID}_{w_i}^S, \mathtt{ID}_{w_{i\to r}})) = H_2(e(\mathtt{ID}_{w_i}, \mathtt{ID}_{w_{i\to r}})^S), \\ S_{ip} = S_{(i,i\to l)} - S_{(i,i\to r)} \bmod n. \end{cases} \tag{6.2}$$

After that, w_i calculate π_i as follows, and sends π_i to $\mathcal{P}$,

$$\begin{aligned}\pi_i &= \bar{C}_i \cdot H_1(t)^{S_{ip}} \bmod n^2 \\ &= g^{\bar{x_i} \cdot y_i} \cdot H_1(t)^{\bar{x_i} \cdot n \cdot \alpha_i} \cdot H_1(t)^{S_{ip}} \bmod n^2.\end{aligned} \tag{6.3}$$

Step 5. $\mathcal{P}$ needs to first check if $\exists\, w_i$ who doesn't submit π_i. If yes, then $\mathcal{P}$ aborts the protocol and starts a new training round. If all the k workers submit their π_i(s), $\mathcal{P}$ calculates $\theta = \prod_{i=1}^{k} \pi_i \bmod n^2$, which can be represented as Eq. (6.4).

$$\theta = g^{\sum_{i=1}^{k} \bar{x_i} \cdot y_i} \cdot H_1(t)^{n \cdot \sum_{i=1}^{k} \bar{x_i} \cdot \alpha_i} \cdot H_1(t)^{\sum_{i=1}^{k} S_{ip}} \bmod n^2. \tag{6.4}$$

Let $\bar{m} = \sum_{i=1}^{k} \bar{x_i} \cdot y_i$, $\mathcal{P}$ can decrypt $\bar{m}$ by $\mathtt{D}(sk, c)$ as

$$\bar{m} = \sum_{i=1}^{k} \bar{x_i} \cdot y_i \bmod n = L(\theta^{\lambda} \bmod n^2) \cdot \mu \bmod n. \tag{6.5}$$

The correctness of Eq. (6.5) is given as follows. First, given $\mathcal{W}_s$, we know that the sum of all the k workers' the processing keys S_{ip} is $\sum_{i=1}^{k} S_{ip} = S_{1p} + S_{2p} + \cdots + S_{kp} = S_{(1,k)} - S_{(1,2)} + S_{(2,1)} - S_{(2,3)} + \cdots + S_{(k,k-1)} - S_{(k,1)} \bmod n$. Based on Eq. (6.2) and the property of Bilinear Pairing, we know $S_{(i,j)} = S_{(j,i)} = H_2(e(\mathtt{ID}_{w_i}, \mathtt{ID}_{w_j})^S)$. Therefore, $\sum_{i=1}^{k} S_{ip} = 0 \bmod n$ as long as $k > 1$ and we can write $\sum_{i=1}^{k} S_{ip} = \gamma \cdot n$ for an integer γ.

Then, Eq. 6.4 can be written as

$$\begin{aligned}\theta &= g^{\sum_{i=1}^{k} \bar{x_i} \cdot y_i} \cdot H_1(t)^{n \cdot \sum_{i=1}^{k} \bar{x_i} \cdot \alpha_i} \cdot H_1(t)^{\gamma \cdot n} \bmod n^2 \\ &= g^{\sum_{i=1}^{k} \bar{x_i} \cdot y_i} \cdot H_1(t)^{n \cdot (\sum_{i=1}^{k} \bar{x_i} \cdot \alpha_i + \gamma)} \bmod n^2.\end{aligned} \tag{6.6}$$

If we consider $H_1(t)^{\sum_{i=1}^{k} \bar{x_i} \cdot \alpha_i + \gamma} \bmod n^2$ as a random number r, $\theta = g^{\bar{m}} \cdot r^n \bmod n^2$ is still a valid Paillier ciphertext and can be decrypted by $\mathtt{D}(sk, c)$. Since m can be either positive (less than $n/2$) or negative (larger than $n/2$), $\mathcal{P}$ can then recover m

by

$$
\begin{cases} m = 10^{-11} \cdot \bar{m}, & \text{if } \bar{m} < \frac{n}{2} \\ m = 10^{-11} \cdot (\bar{m} - n), & \text{otherwise.} \end{cases} \tag{6.7}
$$

Note that, as the original values of x_i and y_i have been respectively amplified by 10^3 and 10^8, the multiplication of 10^{-11} here is to recover the true value of $m = \sum_{i=1}^{k} x_i \cdot y_i$.

After recovering m, $\mathcal{P}$ can update the global model for round $t+1$ by $x^{t+1} \leftarrow \sum_{i=1}^{k} \frac{N_i}{N} x_i = m$. Next, $\mathcal{P}$ will start the next training round by repeating all the above-mentioned steps until the global model achieves a desired performance.

6.4 Security Analysis

In this section, we analyze the security properties of the FedSky scheme. Essentially, given $\forall w_i \in \mathcal{W}_s$ and his/her uploaded model parameter x_i, our analysis will focus on how FedSky is privacy-preserving in protecting each w_i' uploaded model parameter x_i from being disclosed in the whole process of F-MCS. Particularly, based on the assumption that $\mathcal{P}$ does not collude with the workers, $\mathcal{P}$ can get a list of intermediate variables $(\pi_1, \pi_2, \cdots, \pi_k)$ from the k workers where $\pi_i = g^{\bar{x}_i \cdot y_i} \cdot H_1(t)^{\bar{x}_i \cdot n \cdot \alpha_i} \cdot H_1(t)^{S_{ip}} \bmod n^2$ for $i \in [1, k]$. Here, we first prove that the intermediate variable π_i is semantically secure against chosen-plaintext attack (CPA), and then prove the security of our scheme.

6.4.1 The Semantic Security of π_i

Assume that there exist a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ and a hypothetical challenger $\mathcal{C}$. The semantic security of π_i can be proved by a game between $\mathcal{A}$ and $\mathcal{C}$ as follows.

At first, $\mathcal{C}$ runs the key generation algorithm (in Section 6.3.2) to generate $pk_p =$

(n, g), $sk_p = (\lambda, \mu)$ and the hash function H_1. Then $\mathcal{C}$ sends sk_p to $\mathcal{A}$.

Upon receiving sk_p, $\mathcal{A}$ chooses two messages $x_0, x_1 \in \mathbb{Z}_n$, two random numbers $y, \alpha \in \mathbb{Z}_n^*$, and a random integer $t \in [1, 100]$. Then $\mathcal{A}$ sends (x_0, x_1, y, α, t) to $\mathcal{C}$.

After receiving (x_0, x_1, y, α, t), $\mathcal{C}$ first flips a bit $b \in \{0, 1\}$ and generates a random number $S_p \in \mathbb{Z}_n$. Then, $\mathcal{C}$ computes $\pi_b = g^{x_b \cdot y} \cdot H_1(t)^{x_b \cdot n \cdot \alpha} \cdot H_1(t)^{S_p} \bmod n^2$, and returns π_b as a ciphertext back to $\mathcal{A}$.

At last, $\mathcal{A}$ returns $\mathcal{C}$ a bit $b' \in \{0, 1\}$ as the guess of b. As g is the generator of $\mathbb{Z}_{n^2}^*$, there must exist an unique integer $h \in Z_n^*$ such that $H_1(t) = g^h \bmod n^2$ and h is unknown to $\mathcal{A}$. Consequently, π_b can be written as

$$\begin{aligned} \pi_b &= g^{x_b \cdot y} \cdot H_1(t)^{x_b \cdot \alpha \cdot n} \cdot g^{h \cdot S_p} \bmod n^2 \\ &= g^{x_b \cdot y + h \cdot S_p} \cdot H_1(t)^{x_b \cdot \alpha \cdot n} \bmod n^2. \end{aligned} \tag{6.8}$$

Here, π_b can be considered as a valid Paillier ciphertext, thus $\mathcal{A}$ can decrypt $\tilde{m} = (x_b \cdot y + h \cdot S_p) \bmod n$ from π_b.

Next, with (x_0, x_1, y), $\mathcal{A}$ can put x_0 or x_1 into $\tilde{m}$ and calculate $\tilde{m} = x_b \cdot y + h \cdot S_p = x_{1-b} \cdot y + h' \cdot S_p \bmod n$ for $h' \in Z_n^*$. Furthermore, $\mathcal{A}$ knows that h' can be represented as $h' = \frac{\tilde{m} - x_{1-b} \cdot y}{S_p} \bmod n$. However, since S_p is generated by $\mathcal{C}$, $\mathcal{A}$ has no idea about the value of S_p. Therefore, h' is also an unknown and random value to $\mathcal{A}$. So $\mathcal{A}$ cannot distinguish between h' and h. Consequently, $\mathcal{A}$ cannot correctly decide if $b' = 0$ or 1 with a probability higher than 1/2 (i.e., the probability of a random guess). As a result, π_i is semantically secure against $\mathcal{P}$ in the proposed FedSky scheme.

6.4.2 The Security of FedSky

In this section, we first define our scheme's leakage function from the perspective of $\mathcal{P}$. Next, we prove the security of the proposed scheme FedSky in the real/ideal world model.

In the FedSky scheme, the information leaked to $\mathcal{P}$ includes (1) the Paillier public key

pk_p and secret key sk_p, (2) the hash function H_1, (3) the bilinear pairing map e, (4) the list of local dataset sizes $(N_1, N_2, \dots)$ and computational powers $(P_1, P_2, \dots)$, (5) the list of parameter weights $(y_1, y_2, \dots, y_k)$ from the k selected workers, (6) $\bar{m}$ and m where $\bar{m} = \sum_{i=1}^{k} \bar{x}_i \cdot y_i \bmod n$ and $m = \sum_{i=1}^{k} x_i \cdot y_i \bmod n$. Let $\mathtt{L}(\mathcal{P})$ denote the leakage function regarding $\mathcal{P}$, $\mathtt{L}(P) = \{pk_p, sk_p, H_1, e, (N_1, N_2, \dots), (P_1, P_2, \dots), (y_1, y_2, \dots, y_k), \bar{m}, m\}$. Based on $\mathtt{L}(\mathcal{P})$, we define the real/ideal world model as follows.

Real world: In the real world, there exist a probabilistic polynomial time (PPT) adversary $\mathcal{A}$ and a challenger $\mathcal{C}$. $\mathcal{C}$ interacts with $\mathcal{A}$ as follows.

- The $\mathsf{KeyDist}$ Phase: The challenger $\mathcal{C}$ runs the key generation algorithm in Section 6.3.2 to create $pk_p = (n, g), sk_p = (\lambda, \mu)$, the hash function H_1, and the bilinear pairing map e. Next, $\mathcal{C}$ publishes (pk_p, H_1, e), and sends sk_p to $\mathcal{A}$.

- The $\mathsf{WorkSel}$ Phase: $\mathcal{C}$ sends the list of local dataset sizes $(N_1, N_2, \dots)$ and device's computational powers $(P_1, P_2, \dots)$ to $\mathcal{A}$. After that, $\mathcal{A}$ runs the k-point CG-Skyline algorithm to select k workers. Next, $\mathcal{A}$ calculates $C_i = g^{y_i} \cdot H_1(t)^{n \cdot \alpha_i} \bmod n^2$ for each worker w_i where $i \in [1, k]$, and sends $(C_1, C_2, \dots, C_k)$ to $\mathcal{C}$.

- The Challenge Phase: Upon receiving $(C_1, C_2, \dots, C_k)$, $\mathcal{C}$ calculates $(\pi_1, \pi_2, \dots, \pi_k)$ based on the FedSky scheme, and sends $(\pi_1, \pi_2, \dots, \pi_k)$ to $\mathcal{A}$.

- The $\mathsf{DataAgg}$ Phase: After receiving $(\pi_1, \pi_2, \dots, \pi_k)$, $\mathcal{A}$ decrypts $\bar{m} = \sum_{i=1}^{k} \bar{x}_i \cdot y_i \bmod n$ by Eq. (6.6) and calculates $m = \sum_{i=1}^{k} x_i \cdot y_i \bmod n$ based on Eq. (6.7).

Let $\mathtt{View}_{(\mathcal{A}, \mathtt{Real})}$ denote $\mathcal{A}$'s view in the real world. We have $\mathtt{View}_{(\mathcal{A}, \mathtt{Real})} = (\pi_1, \pi_2, \dots, \pi_k)$, where $\pi_i = g^{\bar{x}_i \cdot y_i} \cdot H_1(t)^{\bar{x}_i \cdot n \cdot \alpha} \cdot H_1(t)^{S_{ip}} \bmod n^2$ for $i \in [1, k]$.

Ideal world: In the ideal world, there exist a PPT adversary $\mathcal{A}$ and a simulator $\mathcal{S}$. Besides, $\mathcal{S}$ can get access to the leakage function $\mathtt{L}(P)$. The simulator $\mathcal{S}$ interacts with the adversary $\mathcal{A}$ as follows.

- The $\mathsf{KeyDist}$ Phase: Based on the leaked information in $\mathtt{L}(\mathcal{P})$, $\mathcal{S}$ publishes (pk_p, H_1, e) and sends sk_p to $\mathcal{A}$.

- The $\mathsf{WorkSel}$ Phase: Based on $\mathtt{L}(\mathcal{P})$, $\mathcal{S}$ sends the list of local dataset sizes $(N_1, N_2, \dots)$ and device's computational powers $(P_1, P_2, \dots)$ to $\mathcal{A}$. Then, $\mathcal{A}$ runs the CG-Skyline algorithm to select k workers. Next, $\mathcal{A}$ calculates $C_i = g^{y_i} \cdot H_1(t)^{n \cdot \alpha_i} \bmod n^2$ for each worker w_i where $i \in [1, k]$. At last, $\mathcal{A}$ sends $(C_1, C_2, \dots, C_k)$ to $\mathcal{S}$.

- The Challenge Phase: In this phase, $\mathcal{S}$'s main goal is to generate a list of $(\pi_1', \pi_2', \dots, \pi_k')$ to replace the list of $(\pi_1, \pi_2, \dots, \pi_k)$. Specifically, the list of $(\pi_1', \pi_2', \dots, \pi_k')$ should be generated with random values based on $\mathtt{L}(P)$. Moreover, using $(\pi_1', \pi_2', \dots, \pi_k')$, $\mathcal{A}$ should decrypt the same value of m as using $(\pi_1, \pi_2, \dots, \pi_k)$ in the real world. $\mathcal{S}$ generates $(\pi_1', \pi_2', \dots, \pi_k')$ as follows.

  - At first, $\mathcal{S}$ randomly selects $k-1$ numbers $(x_1', x_2', \dots, x_{k-1}')$ such that $x_i' \in \mathbb{Z}_n^*$ for $i \in [1, k-1]$. Then, given $m = \sum_{i=1}^{k} x_i \cdot y_i \bmod n$ and $(y_1, y_2, \dots, y_k) \in \mathtt{L}(P)$, $\mathcal{S}$ calculates x_k' as:

$$
\begin{aligned}
x_k' &= \frac{(m - \sum_{j=1}^{k-1} x_j' \cdot y_j))}{y_k} \\
&= \frac{(\sum_{i=1}^{k} x_i \cdot y_i - \sum_{j=1}^{k-1} x_j' \cdot y_j)}{y_k} \bmod n.
\end{aligned} \tag{6.9}
$$

    In this case, it is easy to have $m = \sum_{i=1}^{k} x_i \cdot y_i = \sum_{i=1}^{k} x_i' \cdot y_i \bmod n$. Next, $\mathcal{S}$ calculates a list of $(\bar{x_1}', \bar{x_2}', \dots, \bar{x_k}')$, where $\bar{x_i}' = 10^8 \cdot x_i' \bmod n$ for $i \in [1, k]$.

  - $\mathcal{S}$ randomly chooses a sequence of k numbers $(\alpha_1, \alpha_2, \dots, \alpha_k)$ and another sequence of $k-1$ numbers $(S_{1p}', S_{2p}', \dots, S_{(k-1)p}')$ where $\alpha_i, S_{jp}' \in \mathbb{Z}_n^*$ for

$i \in [1, k], j \in [1, k-1]$. Next, $\mathcal{S}$ calculates S'_{kp} as

$$S'_{kp} = -(S'_{1p} + S'_{2p} + \cdots + S'_{(k-1)p}) \bmod n. \tag{6.10}$$

Therefore, it is easy to have $\sum_{i=1}^{k} S'_{ip} = 0 \bmod n$. Hence, we can write $\sum_{i=1}^{k} S'_{ip} = \gamma' \cdot n$ for an integer γ'.

– $\mathcal{S}$ can finally generate the list of $(\pi'_1, \pi'_2, \ldots, \pi'_k)$, where $\pi'_i = g^{\bar{x_i}' \cdot y_i} \cdot H_1(t)^{\bar{x_i}' \cdot \alpha_i \cdot n} \cdot H_1(t)^{S'_{ip}} \bmod n^2$ for $i \in [1, k]$. At last, $\mathcal{S}$ sends $(\pi'_1, \pi'_2, \ldots, \pi'_k)$ to $\mathcal{A}$.

- The $\mathsf{DataAgg}$ Phase: After receiving $(\pi'_1, \pi'_2, \ldots, \pi'_k)$, $\mathcal{A}$ first calculates $\theta' = \prod_{i=1}^{k} \pi'_i \bmod n^2$ as

$$\begin{aligned}
\theta' &= g^{\sum_{i=1}^{k} \bar{x_i}' \cdot y_i} \cdot H_1(t)^{n \cdot \sum_{i=1}^{k} \bar{x_i}' \cdot \alpha_i} \cdot H_1(t)^{\sum_{i=1}^{k} S'_{ip}} \\
&= g^{\sum_{i=1}^{k} \bar{x_i}' \cdot y_i} \cdot H_1(t)^{n \cdot \sum_{i=1}^{k} \bar{x_i}' \cdot \alpha_i} \cdot H_1(t)^{\gamma' \cdot n} \\
&= g^{\sum_{i=1}^{k} \bar{x_i}' \cdot y_i} \cdot H_1(t)^{n \cdot (\sum_{i=1}^{k} \bar{x_i}' \cdot \alpha_i + \gamma')} \bmod n^2.
\end{aligned} \tag{6.11}$$

Similar to the methods in Section 6.3.4, by considering θ' as a valid Paillier ciphertext, $\mathcal{A}$ can decrypt $\bar{m}' = \sum_{i=1}^{k} \bar{x_i}' \cdot y_i \bmod n$. Finally by Eq. (6.7), $\mathcal{A}$ can calculate $m' = \sum_{i=1}^{k} x'_i \cdot y_i = \sum_{i=1}^{k} x_i \cdot y_i \bmod n$, i.e., $\mathcal{A}$ can get the same result as in the real world.

Let $\texttt{View}_{(\mathcal{A},\texttt{Ideal})}$ denote $\mathcal{A}$'s view in the ideal world. We have $\texttt{View}_{(\mathcal{A},\texttt{Ideal})} = (\pi'_1, \pi'_2, \ldots, \pi'_k)$, where $\pi'_i = g^{\bar{x_i}' \cdot y_i} \cdot H_1(t)^{\bar{x_i}' \cdot n \cdot \alpha} \cdot H_1(t)^{S'_{ip}} \bmod n^2$ for $i \in [1, k]$. In Section 6.4.1, we have already proved the semantic security of π_i against CPA. Therefore, $\mathcal{A}$'s view in real world, i.e., $\texttt{View}_{(\mathcal{A},\texttt{Real})} = (\pi_1, \pi_2, \ldots, \pi_k)$, is computationally indistinguishable from its view in the ideal world, i.e., $\texttt{View}_{(\mathcal{A},\texttt{Ideal})} = (\pi'_1, \pi'_2, \ldots, \pi'_k)$. As a result, the proposed FedSky scheme is secure with the leakage function $\texttt{L}(P)$.

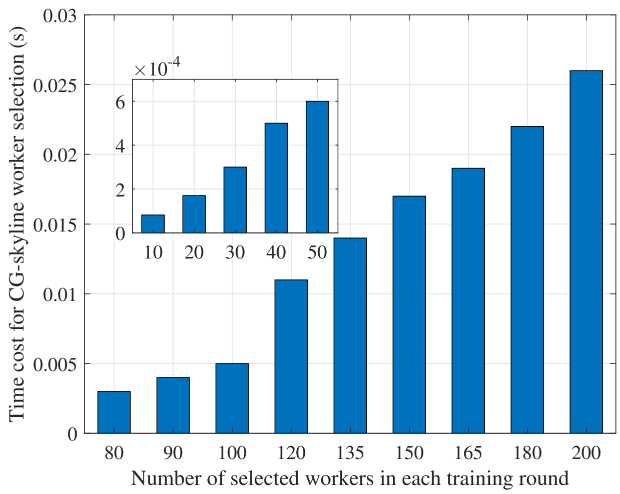
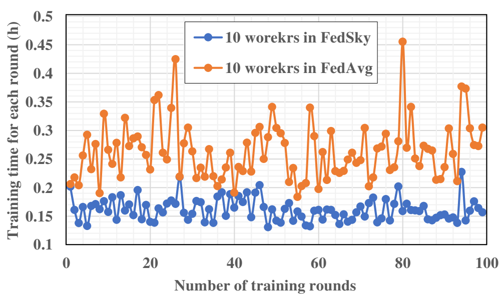
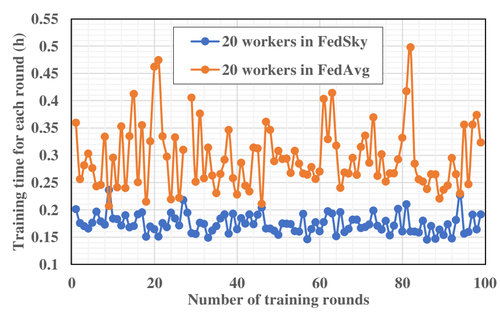


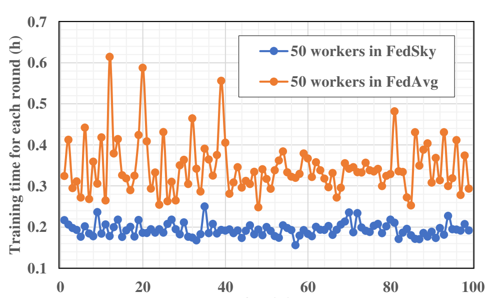
Figure 6.5: The running time for CG-skyline based worker selection with different number of selected workers in each training round.



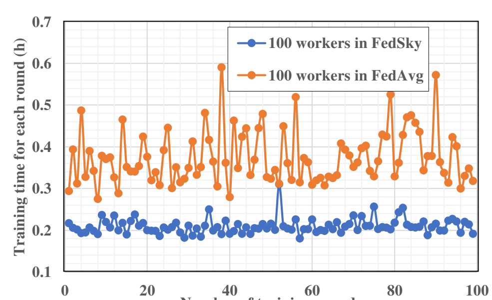
(a) The comparison of workers' local training time (h) per round between FedSky and FedAvg with $sd = 5$ for 10 workers.



(b) The comparison of workers' local training time (h) per round between FedSky and FedAvg with $sd = 5$ for 20 workers.

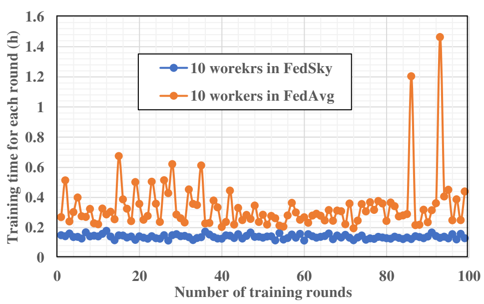


(c) The comparison of workers' local training time (h) per round between FedSky and FedAvg with $sd = 5$ for 50 workers.

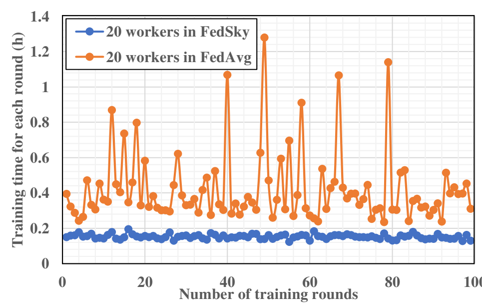


(d) The comparison of workers' local training time (h) per round between FedSky and FedAvg with $sd = 5$ for 100 workers.

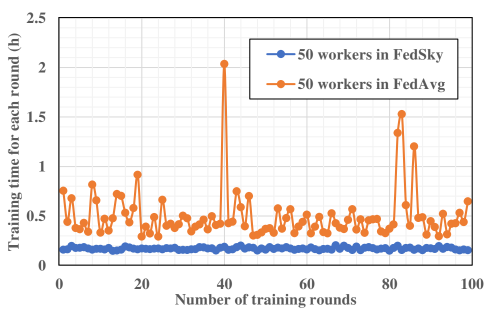
Figure 6.6: The comparison of training time (h) per round between FedSky and FedAvg for different numbers of workers with $sd = 5$.



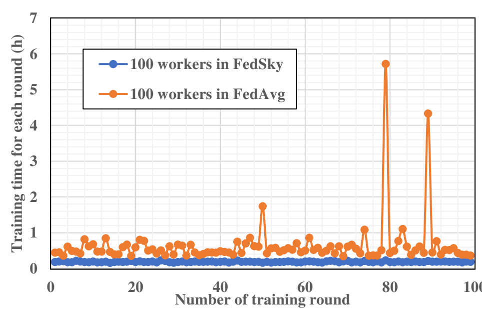
(a) The comparison of workers' local training time (h) per round between FedSky and FedAvg with $sd = 15$ for 10 workers.



(b) The comparison of workers' local training time (h) per round between FedSky and FedAvg with $sd = 15$ for 20 workers.



(c) The comparison of workers' local training time (h) per round between FedSky and FedAvg with $sd = 15$ for 50 workers.



(d) The comparison of workers' local training time (h) per round between FedSky and FedAvg with $sd = 15$ for 100 workers.

Figure 6.7: The comparison of workers' local training time (h) per round between FedSky and FedAvg for different numbers of workers with $sd = 15$.

6.5 Performance Evaluation

In this section, we study the performance of FedSky in terms of time consumption and model accuracy. Particularly, first, we simulated 1000 heterogeneous workers with different computational powers in a F-MCS environment. Then, we compare the time consumption for the FedSky scheme across various experimental settings, including the number of selected workers and the number of training rounds. Second, with the simulated F-MCS environment, we deployed an object classification task on the MNIST dataset with FedSky, and compared the model accuracy between IID data and non-IID data. The details of performance evaluation are presented as follows.

6.5.1 Experimental Setting

In our experiment, we simulated a F-MCS environment including 1000 heterogeneous workers (i.e., $K = 1000$) with different computational powers. We assumed that P_i follows the Gaussian distribution with 50 samples/minute as the mean value and sd the as the standard deviation, where $sd = 5$ or 15 in the experiment.

$\mathcal{P}$ initialized an object classification task on the MNIST dataset [117]. The MNIST dataset is a large database of handwritten digits commonly used for training image processing systems. In the simulation, we assumed that every worker has N_i handwritten digits images, where N_i also follows the Gaussian distribution with 500 as the mean value as 50 as the standard deviation. Then, by following the experimental setup in [77, 106], we designed two ways for distributing the training samples to the workers.

• Independent and identical distribution (IID): There are ten classes (e.g., digits 0 to 9) in the MNIST dataset. In the IID setting, each worker is randomly sampled N_i images across the ten classes from the training dataset.

• Non-Independent and identical distribution (Non-IID): In the Non-IID setting, each worker is still sampled N_i image samples, yet 70% of which come from a dominant class and the remaining 30% belong to other classes. For instance, if $N_i = 500$ for a worker w_i, then she owned 350 images from class "0", while the remaining 150 images are randomly distributed among class "1" to "9".

In each training round, $\mathcal{P}$ selects k workers from 1000 candidates by CG-skyline, where k varies from 10 to 200 in the experiment. There are two constraints $\texttt{Con}_1$ and $\texttt{Con}_2$ for N_i and P_i, respectively. We set $\texttt{Con}_1 = [100, +\infty]$ and $\texttt{Con}_2 = [10, +\infty]$. In the experiment, the security parameter κ is set to 512, i.e., $|p| = |q| = 512$ and $n = 1024$.

Moreover, we deployed a standard convolutional neural network (CNN) as the global model. Particularly, in the CNN model, there are 3 convolution layers with ReLU as the activation function, followed by one fully-connected layer activated by Softmax function. The hyperparameters for training the model are as follows: mini-batch size is 50, number of epochs in each round is 10, the initial learning rate is 0.01 and 0.9 for learning rate decay, number of training rounds is 100. The designed CNN model is implemented using the PyTorch API for the federated learning framework.

All the experiment is implemented with Python programming language conducted experiments on a machine with an Intel(R) Core(TM) i7-6700 CPU @3.60-GHz, 16GB RAM and Windows 10 operating system.

6.5.2 Evaluation Results

In this section, we describe the evaluation results in the experiments. Fig. 6.5 compares the running time (in second) for worker selection based on CG-skyline technique with different k. The CG-skyline is performed in terms of N_i and P_i. For each value of k, the experiment is conducted 10000 times and the average result is reported. From Fig. 6.5, we can see that the running time increases as the increasing

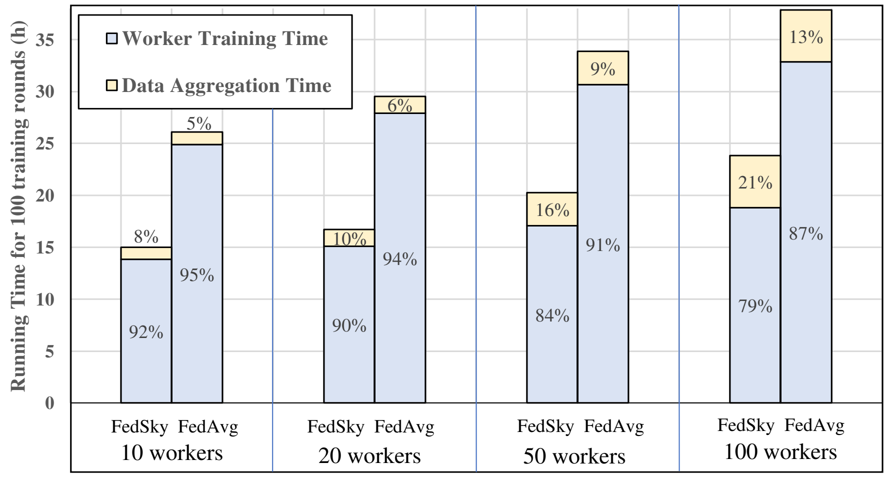
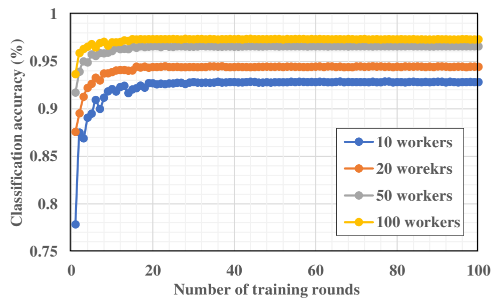


Figure 6.8: Breakdowns of overall running time for 100 training rounds in terms of local training and data aggregation $k = 10, 20, 50$, and 100.

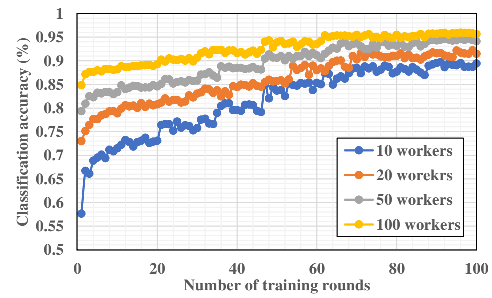
of k. Clearly, worker selection can be finished in a very short period of time (in the order of $10^{-4}s$ to $10^{-2}s$). Compared to the traditional FedAvg scheme, although we add an extra worker selection phase in each training round, the efficiency of the overall process is not affected.

Fig. 6.6 and Fig. 6.7 compare the running time (in hour) for each training round between FedSky and FedAvg for different number of workers with $sd = 5$ and 15, respectively. As mentioned in Section 6.1, we consider the synchronous training protocol, where no worker can proceed to the next training round until all the k workers' data have been uploaded in the current round. Each training round's running time is restricted by the worker who has the poorest computational power. Therefore, in Fig. 6.6 and 6.7, the time required by the worker with the poorest computational power is the running time for each training round.

Notably, the standard deviation sd represents the uncertainty and heterogeneity of workers' computational powers. The larger sd indicates the higher heterogeneity among workers' computational capacities. From Fig. 6.6 and 6.7, we can summarize that, under all the experimental settings, the running time for each training round in FedSky is lower than that in FedSky, indicating the efficiency of FedSky. Specifically,



(a) The accuracy of the deep learning model with different number of workers on IID MNIST data.



(b) The accuracy of the deep learning model with different number of workers on non-IID MNIST data.

Figure 6.9: The performance comparison of the deep learning model between the IID and Non-IID data settings with $k = 10, 20, 50$, and 100.

we can see that for FedAvg, the variance of running time is proportional to both the number of selected workers k and the standard deviation for computational power sd. Under the experimental setting of $k = 100$ and $sd = 15$, the maximum training time for FedSky can be as high as 6 hours. On the other hand, we notice that neither the number of selected workers k nor the standard deviation sd can greatly affect the performance of FedSky. For all the experimental settings, FedSky takes about 0.2 hours to finish one training round, indicating the reliability and stability of FedSky in terms of time consumption. The performance difference between FedSky and FedAvg is even more significant as the increase of k and sd.

In Fig. 6.8, we compare the breakdowns of the overall running time for 100 training rounds in terms of workers' local training time and the platform's aggregation time. Same as [127], we found that for both FedSky and FedAvg, workers' local training time dominates the overall time consumption of the FL process. Especially in FedAvg, the F-MCS platform can perform data aggregation very quickly. However, most of its time is wasted on waiting for the last worker to send the encrypted model updates. For instance, in FedAvg, the actual computation time for the platform $\mathcal{P}$ to aggregate and decrypt data only accounts for 5% to 13% of the whole time span. In contrast, FedSky can greatly increase workers' training efficiency and reduce $\mathcal{P}$'s idle time. For instance, when $k = 100$, $\mathcal{P}$'s idle time will be decreased from 87% (in FedAvg) to 79% (in FedSky). Overall speaking, by introducing an extra worker selection phase, FedSky can select qualified workers in terms of local data size and computational power. The time consumption of worker selection in FedSky is quite low (in the order of $10^{-2}s$ to $10^{-4}s$). But the overall efficiency of the training process can be greatly improved.

Fig. 6.9 shows the comparison of model performance with the different numbers of selected workers for both IID and Non-IID settings. We can see that for both cases, selecting more workers can lead to better classification accuracies. Besides, we notice that when $k = 100$, it takes only about ten rounds to achieve 96% target accuracy in the IID setting. However, it takes about 100 rounds to achieve similar accuracy in the Non-IID setting. Notably, data samples across mobile workers are usually Non-IID in the real-world F-MCS scenarios. Hence, the F-MCS service needs more training rounds to achieve a satisfactory result.

6.6 Related Work

6.6.1 Worker Selection in MCS

Worker selection is always a fundamental problem in MCS applications. Tong et al. [104] proposed a micro-task allocation problem in spatial crowdsourcing. In their study, they focused on the simple tasks any individual can perform with the smart devices. Specifically, they used the online maximum weighted bipartite matching problem as the baseline algorithm, and introduced a two-phase-based framework to improve the baseline's efficiency. Zhao et al. [142] studied a destination-aware task assignment problem that concerns the optimal strategy of assigning each task to proper workers. Consequently, the total number of completed tasks can be maximized while all workers can reach their destinations before deadlines after performing assigned tasks. Zhao et al. [143] focused on a novel spatial crowdsourcing problem, namely predictive task assignment, which aimed to maximize the number of assigned tasks by considering both current and future workers in a dynamic way. Zhang et al. [137] proposed a privacy-preserving worker selection scheme for MCS platform based on probabilistic skyline query over sliding windows. Their work can continuously select reliable workers in terms of working experience, expiry time, and trustability without revealing workers' sensitive information. Unlike the above studies, in this chapter, we considered the worker selection problem under the F-MCS circumstance where the task is complicated (e.g., image classification or object detection) and the model needs to be trained across workers' local datasets.

6.6.2 Privacy-preservation FL

Zhang et al. [127] proposed a homomorphic encryption-based scheme for *cross-silo* FL. Instead of encrypting individual gradients, they encoded a batch of quantized gradients into a long integer and encrypted it in one go. The experimental results showed that the proposed method can significantly increase the training speedup while reducing the communication overhead. Li et al. [54] designed a *cross-silo* FL scheme to detect cyber threats in industrial cyber-physical systems based on Paillier cryptosystem. Their system model contains one cloud server and multiple physical infrastructures (i.e., workers). The cloud server aggregates the encrypted model updates and sends the encrypted model to the workers. Each worker holds the private key and decrypts the model to build a local intrusion detection system. Mandal et al. [69] presented a privacy-preserving system for training linear and logistic regression models. Their proposed model can guarantee data and model privacy as well as ensuring robustness to users dropping out in the network. The core of their training protocols is a secure multiparty global gradient computation on alive users' data. Their security goal is to protect the trained model from being leaked to the user, but we do not guarantee this assumption in our work. Overall, most of the existing solutions focus on privacy-preserving data aggregation in *cross-silo* FL system for different applications.

6.7 Concluding Remarks

In this chapter, we have proposed a privacy-preserving scheme for F-MCS applications, called FedSky. By extending the classic FedAvg algorithm, FedSky selects qualified workers based on CG-skyline technique and securely aggregate model updates for training the global model. Particularly, compared with FedAvg, our scheme considers the workers' dynamics and heterogeneous natures. It can significantly improve the efficiency of model training process in F-MCS. Besides, we have designed a novel privacy-preserving data aggregation protocol based on the additive HE system. This protocol is exclusively designed for *cross-device* FL setting and requires no interactions between workers during the model training process. Security analysis demonstrates that the proposed scheme is privacy-preserving. Extensive experiments have been conducted on a real-world image classification task. The comparison results validate the efficiency of worker selection and the robustness of FedSky against heterogeneous workers.

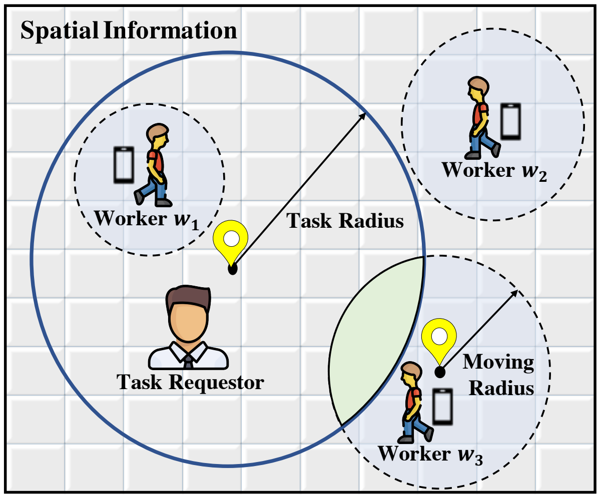
Chapter 7

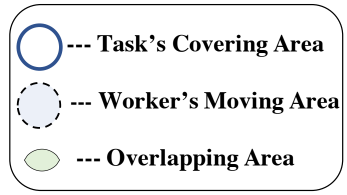
Spatial-temporal Similarity based Privacy-preserving Worker Selection in MCS

7.1 Introduction

Up until now, we have proposed three privacy-preserving schemes for solving three major challenges in MCS worker selection, namely the evaluation of workers' trustability, dynamic worker selection, and the protection of workers' sensing data regarding their heterogeneous characteristics. In the following chapters, we will mainly focus on privacy-preserving worker selection in terms of spatial-temporal matching. As we discussed, MCS is a newly-emerged sensing paradigm where a group of workers is selected to collect and share real-time data for a particular task [119]. For example, the local government wants to recruit mobile workers to observe the local air quality in an industry zone during the daytime; a real estate manager wants to hire mobile workers to collect the noisy levels in a local community at night. In such cases, the task requestor (i.e., the government or the real estate manager) needs to define the spatial-temporal information, including the task covering area and the time duration. Usually, the task's coverage area is relatively big (e.g., a local community), and the task's time duration is pretty long (e.g., 8:00 am to 5:00 pm). Therefore, it is not possible for a single worker to cover the entire task area throughout the time duration. Consequently, based on each worker's moving area and the corresponding time duration in the area, the MCS platform should select worker(s) who has a high spatial-temporal similarity with the sensing task.

Fig. 7.1 shows a motivation example. There are three workers in the system, w_1, w_2 and w_3. Each worker has his/her moving area and time duration. A task requestor launches a MCS task with a given task covering area (i.e., the big circular area) and time duration (8:00 am to 12:00 pm and 13:00 pm to 16:00 pm). From Fig. 7.1, we can see that w_1's moving area is totally enclosed in the task coverage area, but their overlapping time duration is quite short (i.e., only 1 hour). For worker w_2, he/she has no spatial intersection with the task at all. For worker w_3, although his/her overlapping time duration with the task is long (i.e., totally 5 hours), there is a smaller overlapping area than w_1. There are two major challenges for the MCS platform to select workers in such scenario. The first challenge is that the MCS task's budget is always limited, so not all the workers can be selected to perform the task. Therefore, the MCS platform needs to accurately calculate the spatial-temporal similarity between the sensing task and each worker and only selects the workers with high spatial-temporal similarities. The second challenge is that, in worker selection, it is inevitable for the task requestor and the workers to share some of their sensitive spatial-temporal information, e.g., the moving area and the corresponding time duration. Both the requestor and the workers may be reluctant to participate in the task due to the potential privacy disclosure. Therefore, the MCS platform should protect their spatial-temporal information while still select proper workers effectively.





Temporal Information

	Time Duration
Task	8:00-12:00 **AND** 13:00-16:00
w_1	10:00 − 11:00
w_2	9:00-11:00 **AND** 15:00-16:00
w_3	8:00-13:00 **AND** 15:00-16:00

Figure 7.1: A motivation example of a MCS platform with one task requestor, three workers, and their corresponding spatial-temporal information.

Recently, numerous studies focused on worker selection in MCS applications [136, 132, 76, 102]. However, very few of them simultaneously address the two challenges we mentioned above. Aiming to fill this research gap, we study the problem of privacy-preserving worker selection in terms of the spatial-temporal similarity between the task and the workers. The overall contributions of this worker are threefold as follows.

- First, we propose a novel worker selection scheme in MCS services, where the MCS platform can select qualified workers who have high spatial-temporal similarity with the sensing task. Specifically, we provide the detailed definition and calculation of spatial-temporal similarity in MCS services.

- Second, we design a novel privacy-preserving approach to protect participants' spatial-temporal information based on the modified Paillier encryption technique. In the proposed scheme, each worker does not need to provide his/her moving area and time duration directly, while the cloud servers can still select the proper workers effectively.

- Third, Security analysis shows that the proposed scheme can effectively pre-

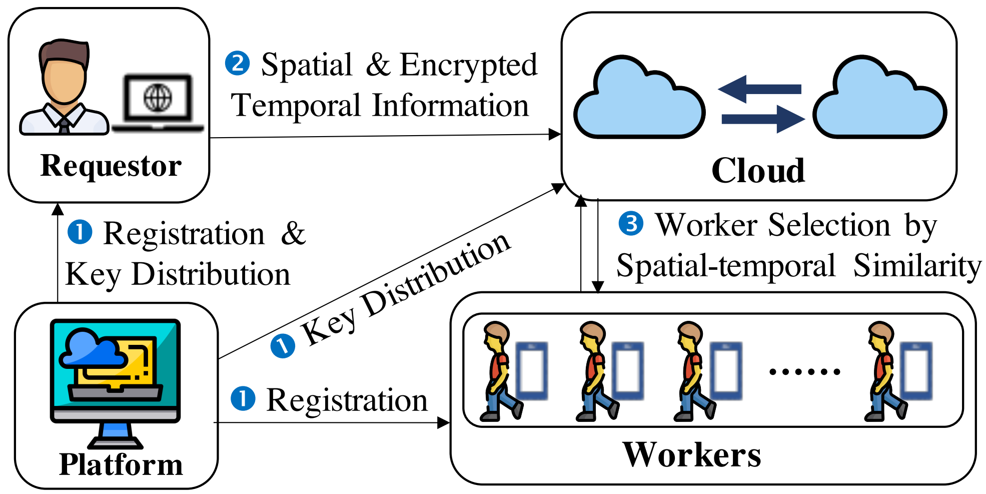


Figure 7.2: The system model under consideration in chapter 7.

serve the privacy of the participants' sensitive information. In addition, we build a custom simulator for performance evaluation. The experimental results validate the effectiveness and trustability of the proposed scheme.

The remainder of this chapter is organized as follows. In Section 7.2, we introduce our system model, security model and design goals. In Section 7.3, we discuss the definition of spatial-temporal similarity between workers and tasks. In Section 7.4 we propose our scheme in details. Then in Section 7.5 and Section 7.6, we present the security analysis and performance evaluation, followed by the related work in Section 7.7. Finally, we recap the conclusions in Section 7.8.

7.2 Models and Design Goals

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

7.2.1 System Model

Our system model (see in Fig. 7.2) mainly consists of four entities, namely, a MCS platform $\mathcal{P}$, a task requestor $\mathcal{R}$, a cloud platform with two servers $\{\mathcal{S}_1, \mathcal{S}_2\}$, and a group of participating workers $\mathcal{W} = \{w_1, w_2, \dots\}$.

• The MCS Platform $\mathcal{P}$: $\mathcal{P}$ is the service organizer who is responsible for bootstrapping the entire MCS system. Mainly, $\mathcal{P}$ registers task requestor and workers before the sensing tasks start. Besides, $\mathcal{P}$ generates and distributes keys to different entities, so that a certain task can be completed cooperatively. After that, $\mathcal{P}$ just stays offline.

• Cloud Servers $\{\mathcal{S}_1, \mathcal{S}_2\}$: There are two cloud servers $\{S_1, S_2\}$ in our system model. Essentially, S_1 and S_2 will work together to select qualified workers based on the spatial-temporal similarity between the workers and the sensing task. After worker selection, $\{\mathcal{S}_1, \mathcal{S}_2\}$ collects the sensing data from the workers, and then sends the data to the task requestor.

• Task Requestor $\mathcal{R}$: In MCS services, $\mathcal{R}$ is the task initiator (e.g., government, companies, or individuals). At first, $\mathcal{R}$ needs to define the spatial coverage area and the time duration for the task, i.e., the task's spatial-temporal information. Next, $\mathcal{R}$ needs to send the spatial-temporal information to the cloud servers $\{\mathcal{S}_1, \mathcal{S}_2\}$ for worker selection.

• Workers $\mathcal{W}$: $\mathcal{W} = \{w_1, w_2, \dots w_d\}$ are d participating workers who wish to conduct the MCS tasks. Each worker randomly walks around in a specific local area for a certain period of time and collects real-time sensing data. In this work, only the workers with high spatial-temporal similarity with the sensing task will be selected.

7.2.2 Security Model

In our security model, we consider $\mathcal{P}$ and $\mathcal{R}$ are trustable. However, $\{\mathcal{S}_1, \mathcal{S}_2\}$ are semi-honest, which means both of them strictly follow the protocol, yet may be curious to learn $\mathcal{R}$ or $\mathcal{W}$'s spatial-temporal information. For $\mathcal{W}$, we assume that they are strategic and selfish for maximizing their own profits. Hence, if a worker provides dishonest information, he/she may have a higher chance of being selected.

However, in this work, we deploy the reputation-based evaluation system to assess and update workers' trustabilities [136]. Only the workers with high trustabilities will be considered. Therefore, from a long-term perspective, each worker should submit their personal spatial-temporal information as accurately as possible. Besides, there is no collusion between $\mathcal{S}_1$ and $\mathcal{S}_2$, and the task requestor $\mathcal{R}$/workers cannot collude with the two cloud servers either.

It is worth noting that outside attackers may exploit other vulnerabilities of the MCS system, as this work focuses on privacy-preservation, those attack are beyond the scope of this thesis and will be discussed in our future work.

7.2.3 Design Goals

Our goal in this chapter is to select reliable and suitable workers for conducting a sensing task. Notably, the following objective should be satisfied.

• For the MCS task, each task requestor $\mathcal{R}$ is required to report the task's coverage area $\mathtt{Area_t}$ and the task's time duration $\tau_{\mathtt{t}}$. In our security model, we consider $\mathtt{Area_t}$ is non-sensitive information. This is because in a real-world MCS scenario, $\mathtt{Area_t}$ is usually a relative big and public area, such as a local community, the campus of an university, or a shopping district. By only knowing $\mathtt{Area_t}$, $\mathcal{S}_1$ and $\mathcal{S}_2$ can never know the requestor's daily behaviour or exact real-time location. However, we consider $\tau_{\mathtt{t}}$ is $\mathcal{R}$' sensitive information and should be protected from being disclosed. If $\tau_{\mathtt{t}}$ is leaked during the worker selection, then $\{\mathcal{S}_1, \mathcal{S}_2\}$ can easily link $\tau_{\mathtt{t}}$ with $\mathtt{Area_t}$ and guess $\mathcal{R}$' task purposes. As a result, after many tasks, $\{\mathcal{S}_1, \mathcal{S}_2\}$ can infer $\mathcal{R}$' daily behaviours and routines by analysing the task purposes.

• For a worker w_i, we consider both w_i's moving area $\mathtt{Area}_{w_i}$, the time duration τ_{w_i}, and the spatial-temporal similarity $\mathtt{Sim}_{(\mathtt{t},w_i)}$ are sensitive information and should be protected from being disclosed. This is because the moving area $\mathtt{Area}_{w_i}$ is always a relative small area, in which a person can be easily tracked. Besides, to be selected in the MCS services, w_i need to report his/her $\texttt{Area}_{w_i}$ and τ_{w_i} very frequently. It is very easy for $\{\mathcal{S}_1, \mathcal{S}_2\}$ to rebuild w_i's real-time locations and daily path by tracing the spatial-temporal similarity $\texttt{Sim}_{(\texttt{t},w_i)}$.

In summary, our goal is to select qualified workers in the MCS platform while protecting the privacy of the sensing task's time duration $\tau_{\texttt{t}}$ and w_i's spatial-temporal information $\texttt{Area}_{w_i}$, τ_{w_i} and $\texttt{Sim}_{(\texttt{t},w_i)}$.

7.3 Spatial-temporal Similarity between Works and the Task

This section presents the spatial-temporal similarity, which is proposed in [138] as contact similarity between trajectories.

Definition 7.3.1. *A* ***MCS task*** $\mathcal{T}$ *is defined by the task requestor $\mathcal{R}$, which contains the task's covering area $\texttt{Area}_t$ and time duration τ_t. For simplicity sake, in this work, $\texttt{Area}_t$ is considered as a circle centering at (x_t, y_t) with a given radius R_t. Hence, the equation of $\texttt{Area}_t$ can be represented as*

$$(x - x_t)^2 + (y - y_t)^2 = R_t^2. \tag{7.1}$$

Besides, τ_t is the time duration for the MCS task $\mathcal{T}$. Notably, τ_t does not have to be a continuous time period. It can be a combination of several time periods. For example, in the example of Fig. 7.1, $\tau_t = (8:00, 12:00) \cap (13:00, 16:00)$.

Definition 7.3.2. *A MCS worker w_i'* ***spatial-temporal information*** *consists of his/her moving area $\texttt{Area}_{w_i}$ and time duration τ_{w_i} that w_i is in this area. In this work, we assume that each worker randomly moves around in $\texttt{Area}_{w_i}$ and collects the real-time MCS data during τ_{w_i}. Again, for simplicity sake, we consider $\texttt{Area}_{w_i}$*

is a circle centered at (x_{w_i}, y_{w_i}) *with radius* R_{w_i}*. Hence, the equation for* $\texttt{Area}_{w_i}$ *is*

$$(x - x_{w_i})^2 + (y - y_{w_i})^2 = R_{w_i}^2. \tag{7.2}$$

Definition 7.3.3. *Given a MCS task* $\mathcal{T}$*'s covering area* $\texttt{Area}_t$ *and a worker* w_i*'s moving area* $\texttt{Area}_{w_i}$*, if* $\texttt{Area}_t$ *spatial overlaps with* $\texttt{Area}_{w_i}$ *(denoted as* $\texttt{Area}_t \cap \texttt{Area}_{w_i} \neq \emptyset$*), then the* ***intersection area ratio*** $R_{(t,w_i)}$ *between* $\mathcal{T}$ *and* w_i *is*

$$R_{(t,w_i)} = \texttt{Area}_{(t,w_i)}/\texttt{Area}_t, \tag{7.3}$$

where $\texttt{Area}_{(t,w_i)}$ *is the intersection area between the* $\texttt{Area}_t$ *and* $\texttt{Area}_{w_i}$*. In this work, we assume that* $0 < R_{(t,w_i)} < 1$*.* $\texttt{Area}_t > \texttt{Area}_{w_i}$ *for all the time, so* $0 < R_{(t,w_i)} < 1$*.*

Definition 7.3.4. *Given a MCS task* $\mathcal{T}$*'s time duration* τ_t *and a worker* w_i*'s time duration* τ_{w_i}*, the* ***overlapping time duration*** $\tau_{(t,w_i)}$ *between* w_i *and* $\mathcal{T}$ *can be calculated as*

$$\tau_{(t,w_i)} = \sum_{\tau \subset \tau_t \wedge \tau_{w_i}} \tau. \tag{7.4}$$

Example 7.3.1. *Let consider the example in Fig. 7.1 where* $\tau_t = (8:00, 12:00) \cap (13:00, 16:00)$ *and* $\tau_{w_1} = (10:00, 11:00)$*. We can simply get* $\tau_{(t,w_1)} = (10:00, 11:00) = 1$ *hour.*

Definition 7.3.5. *Given a MCS task* $\mathcal{T}$ *and a worker* w_i*, the* ***spatial-temporal similarity*** $\texttt{Sim}_{(t,w_i)}$ *between worker* w_i *and the sensing task* $\mathcal{T}$ *can be calculated as [138]:*

$$\texttt{Sim}_{(t,w_i)} = 1 - (1 - R_{(t,w_i)})^{\tau_{(t,w_i)}}. \tag{7.5}$$

We assume that $\mathtt{R}_{(t,w_i)}$ indicates the probability that w_i spatially overlaps with the task $\mathcal{T}$ per unit time. Therefore, $(1 - \mathtt{R}_{(t,w_i)})^{\tau_{(t,w_i)}}$ denotes the probability that w_i does not spatially overlaps with $\mathcal{T}$ during the period time $\tau_{(t,w_i)}$. Consequently,

	Time Duration
Task	8: 00-12: 00 **AND** 13: 00-16: 00
w_1	10: 00-11: 00
w_2	9: 00-11: 00 **AND** 15: 00-16:00
w_3	8: 00-13: 00 **AND** 15: 00-16: 00

(a) Time duration for the task and workers

Time Slots	Time Slot τ_1 8: 00-9: 00	Time Slot τ_2 9: 00-10: 00	Time Slot τ_3 10: 00-11: 00	Time Slot τ_4 11: 00-12:00	Time Slot τ_5 12: 00-13:00	Time Slot τ_6 13: 00-14:00	Time Slot τ_7 14: 00-15:00	Time Slot τ_8 15: 00-16:00
Task	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8
w_1	0	0	1	0	0	0	0	0
w_2	0	1	1	1	0	0	0	1
w_3	1	1	1	1	1	0	0	1

(b) Overlapping time duration calculation table between the task and workers

Figure 7.3: An illustration of calculating the overlapping time duration between the task and three workers.

$1-(1-\mathtt{R}_{(\mathtt{t},w_i)})^{\tau_{(\mathtt{t},w_i)}}$ means the probability that w_i spatially overlaps with $\mathcal{T}$ for time duration $\tau_{(\mathtt{t},w_i)}$, i.e., the spatial-temporal similarity between w_i and $\mathcal{T}$. More details about the contact similarity can be found in Appendix A.

7.4 Our Proposed Scheme

In this work, $\mathcal{P}$ securely selects workers based on the spatial-temporal similarity $\mathtt{Sim}_{(\mathtt{t},w_i)}$. Given the fact that, $f(\mathtt{t},w_i) = (1-\mathtt{R}_{(\mathtt{t},w_i)})^{\tau_{(\mathtt{t},w_i)}}$ has the same monotonicity with $\hat{f}(\mathtt{t},w_i) = \tau_{(\mathtt{t},w_i)} \cdot \ln(1-\mathtt{R}_{(\mathtt{t},w_i)})$, we design our scheme with the four phases as follows.

7.4.1 The System Initialization Phase ($\mathtt{SysInit}$)

At first, $\mathcal{P}$ runs $\mathsf{Paillier.KeyGen}(\kappa)$ to generate public key $pk = (n, g, h = g^s \mod n^2)$ and secret key $sk = s$. Next, $\mathcal{P}$ generates s_1 and s_2 such that $s_1 + s_2 = s$. After that, $\mathcal{P}$ distributes pk to the task requestor $\mathcal{R}$ and workers $\mathcal{W}$, and then distributes s_1 and s_2 to $\mathcal{S}_1$ and $\mathcal{S}_2$, respectively. Besides, $\mathcal{P}$ splits a day into v different time slots, i.e., $\tau_{\mathtt{slot}} = \{\tau_1, \tau_2, \ldots, \tau_v\}$. For example, if $v = 24$, then each time slot represents 1 hour time duration. Then, $\mathcal{P}$ creates a table with v columns, in which each column indicates if the corresponding time slot is covered by $\tau_{\mathtt{t}}$ or τ_{w_i}. This table will be distributed to both $\mathcal{R}$ and $\mathcal{W}$, and will be used for overlapping time duration calculation. After initializing the entire system, $\mathcal{P}$ just stays offline and will not participate in the MCS services.

7.4.2 The Task Launching Phase ($\mathtt{TaskLun}$)

For each time slot $\tau_j \in \tau_{\mathtt{slot}}$, $\mathcal{R}$ first chooses a random number $r_j \in [1, n/4]$, and then given $pk = (n, g, h)$, $\mathcal{R}$ calculates $c_j = (C_{(j,1)}, C_{(j,2)})$, where

$$c_j = \begin{cases} C_{(j,1)} = g^{r_j} \bmod n^2, \\ C_{(j,2)} = h^{r_j}(1 + m_j \cdot n) \bmod n^2. \end{cases} \tag{7.6}$$

Here, $m_j = 1$ if $\tau_j \subset \tau_{\mathtt{t}}$, and $m_j = 0$ otherwise. Next, $\mathcal{R}$ adds each c_j into the j-th column in the overlapping time duration calculation table (see the first two rows in Fig. 7.3 (b)). Finally, $\mathcal{R}$ sends $\mathtt{Area_t}$ and the overlapping time duration calculation table to $\mathcal{S}_1$. After that, $\mathcal{S}_1$ distributes $\mathtt{Area_t}$ and the table to all the active workers who want to perform the sensing task.

7.4.3 The Local Calculation Phase ($\mathtt{LolCal}$)

For worker w_i, after receiving $\mathtt{Area_t}$ and the overlapping time duration table, he/she will first calculate the intersection area $\mathtt{Area}_{(\mathtt{t},w_i)}$. We need to calculate the intersection area $\mathtt{Area}_{(\mathtt{t},w_i)}$ for the following two cases (as see in Fig. 7.4). In the first case (e.g., w_1), $\mathtt{Area}_{w_i}$ is totally enclosed in $\mathtt{Area_t}$, hence $\mathtt{Area}_{(\mathtt{t},w_i)} = \mathtt{Area}_{w_i} = \pi \cdot R_{w_i}^2$. In the second case (e.g., w_2), $\mathtt{Area}_{(\mathtt{t},w_i)} = A_1 + A_2$, where

$$\begin{cases} A_1 = 2\int_{d_1}^{R_{\mathtt{t}}} \sqrt{R_{\mathtt{t}}^2 - x^2}dx, \\ A_2 = 2\int_{d-R_{w_2}}^{d_1} \sqrt{R_{w_i}^2 - (x-d)^2}dx. \end{cases} \tag{7.7}$$

Here, $d = \sqrt{(x_{\mathtt{t}} - x_{w_i})^2 + (y_{\mathtt{t}} - y_{w_i})^2}, d_1 = \frac{R_{\mathtt{t}}^2 - R_{w_i}^2 + d^2}{2d}$. After that, if $\mathtt{Area}_{(\mathtt{t},w_i)} \neq 0$, w_i will calculate β_i as

$$\beta_i = \lfloor 10^5 \cdot \ln(1 - \mathtt{R}_{(\mathtt{t},w_i)}) \rfloor = \lfloor 10^5 \cdot \ln(1 - \mathtt{Area}_{(\mathtt{t},w_i)}/\mathtt{Area_t}) \rfloor. \tag{7.8}$$

Here, we retain $\ln(1-\mathtt{R}_{(\mathtt{t},w_i)})$ to 5 decimal places, and 10^5 is a scaling coefficient which converts $\ln(1-\mathtt{R}_{(\mathtt{t},w_i)})$ to a positive integer in $\mathbb{Z}_n$. Next, w_i generates a time slot index set $\mathbb{C}_{w_i}$ such that $\mathbb{C}_{w_i} = \{j|\tau_j \subset \tau_{w_i}\}$. In the example of Fig. 7.3 (a), $\mathbb{C}_{w_1} = \{3\}$, $\mathbb{C}_{w_2} = \{2,3,4,8\}$, and $\mathbb{C}_{w_3} = \{1,2,3,4,5,8\}$. Then, w_i calculates $c_{w_i} = (C_{(w_i,1)}, C_{(w_i,2)})$, where $C_{(w_i,1)} = \prod_{j\in\mathbb{C}_{w_i}} C_{(j,1)}$ and $C_{(w_i,2)} = \prod_{j\in\mathbb{C}_{w_i}} C_{(j,2)}$, i.e.,

$$\begin{cases} C_{(w_i,1)} = g^{\sum_{j\in\mathbb{C}_{w_i}} r_j} \bmod n^2, \\ C_{(w_i,2)} = h^{\sum_{j\in\mathbb{C}_{w_i}} r_j} \cdot \prod_{j\in\mathbb{C}_{w_i}} (1+m_j\cdot n) \bmod n^2. \end{cases} \tag{7.9}$$

After that, w_i calculates $\widetilde{c}_{w_i} = \{\widetilde{C}_{(w_i,1)}, \widetilde{C}_{(w_i,2)}\}$ where $\widetilde{C}_{(w_i,1)} = C^{\beta_i}_{(w_i,1)}$ and $\widetilde{C}_{(w_i,2)} = C^{\beta_i}_{(w_i,2)}$. So we have

$$\begin{cases} \widetilde{C}_{(w_i,1)} = g^{\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j} \bmod n^2, \\ \widetilde{C}_{(w_i,2)} = h^{\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j} \cdot \prod_{j\in\mathbb{C}_{w_i}} (1+m_j\cdot n)^{\beta_i} \bmod n^2. \end{cases} \tag{7.10}$$

Finally, w_i sends $\widetilde{c}_{w_i} = \{\widetilde{C}_{(w_i,1)}, \widetilde{C}_{(w_i,2)}\}$ to the cloud $\mathcal{S}_1$.

7.4.4 The Similarity Comparison Phase (SimComp)

For each sensing task, $\mathcal{S}_1$ first randomly chooses two numbers $a, b \in [1, n/4]$, which can be used to protect workers' similarities from being decrypted by $\mathcal{S}_2$. After receiving $\widetilde{c}_{w_i}$ from w_i, $\mathcal{S}_1$ then calculates $c_a = \{C_{(a,1)}, C_{(a,2)}\}$ where $C_{(a,1)} = \widetilde{C}^a_{(w_i,1)}$ and $C_{(a,2)} = \widetilde{C}^a_{(w_i,2)}$, i.e.,

$$\begin{cases} C_{(a,1)} = g^{a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j} \bmod n^2, \\ C_{(a,2)} = h^{a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j} \cdot \prod_{j\in\mathbb{C}_{w_i}} (1+m_j\cdot n)^{a\cdot\beta_i} \bmod n^2. \end{cases} \tag{7.11}$$

After that, $\mathcal{S}_1$ chooses a random number $r_i \in [1, n/4]$ and calculates $c_b = \{C_{(b,1)}, C_{(b,2)}\}$, such that

$$\begin{cases} C_{(b,1)} = g^{r_i} \bmod n^2 \\ C_{(b,2)} = h^{r_i}(1 + b \cdot n) \bmod n^2. \end{cases} \tag{7.12}$$

Moreover, $\mathcal{S}_1$ computes $c_{ab} = \{C_{(ab,1)}, C_{(ab,2)}\}$ such that $C_{(ab,1)} = C_{(a,1)} \cdot C_{(b,1)}$ and $C_{(ab,2)} = C_{(a,2)} \cdot C_{(b,2)}$. Here, let $\Theta = \prod_{j \in \mathbb{C}_{w_i}} (1 + m_j \cdot n)^{a \cdot \beta_i} \bmod n^2$, we have

$$\begin{cases} C_{(ab,1)} = g^{a \cdot \beta_i \cdot \sum_{j \in \mathbb{C}_{w_i}} r_j + r_i} \bmod n^2, \\ C_{(ab,2)} = h^{a \cdot \beta_i \cdot \sum_{j \in \mathbb{C}_{w_i}} r_j + r_i} \cdot \Theta \cdot (1 + b \cdot n) \bmod n^2. \end{cases} \tag{7.13}$$

Next, $\mathcal{S}_1$ partially decrypts c_{ab} by computing $\widetilde{c}^*_{w_i} = \{\widetilde{C}^*_{(w_i,1)}, \widetilde{C}^*_{(w_i,2)}\}$ based on modified Paillier PKE. So we have

$$\begin{cases} \widetilde{C}^*_{(w_i,1)} = \widetilde{C}_{(ab,1)} = g^{a \cdot \beta_i \cdot \sum_{j \in \mathbb{C}_{w_i}} r_j + r_i} \bmod n^2, \\ \widetilde{C}^*_{(w_i,2)} = \frac{\widetilde{C}_{(ab,2)}}{(\widetilde{C}_{(ab,1)})^{s_1}} = \frac{h^{a \cdot \beta_i \cdot \sum_{j \in \mathbb{C}_{w_i}} r_j + r_i} \cdot \Theta \cdot (1 + b \cdot n)}{g^{s_1 \cdot (a \cdot \beta_i \cdot \sum_{j \in \mathbb{C}_{w_i}} r_j + r_i)}} \bmod n^2. \end{cases} \tag{7.14}$$

After that, $\mathcal{S}_1$ sends $\widetilde{c}^*_{w_i} = \{\widetilde{C}^*_{(w_i,1)}, \widetilde{C}^*_{(w_i,2)}\}$ to $\mathcal{S}_2$. Upon receiving $\widetilde{c}^*_{w_i}$, $\mathcal{S}_2$ computes

$M_{w_i} = \frac{\widetilde{C}^*_{(w_i,2)}}{(\widetilde{C}^*_{(w_i,1)})^{s_2}}$, such that

$$\begin{aligned}
M_{w_i} &= \frac{h^{a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j + r_i}\cdot\Theta\cdot(1+b\cdot n)}{g^{s_1\cdot(a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j + r_i)}\cdot g^{s_2\cdot(a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j + r_i)}} \bmod n^2 \\
&= \frac{h^{a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j + r_i}\cdot\Theta\cdot(1+b\cdot n)}{g^{(s_1+s_2)\cdot(a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j + r_i)}} \bmod n^2 \\
&= \frac{h^{a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j + r_i}\cdot\Theta\cdot(1+b\cdot n)}{g^{s\cdot(a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j + r_i)}} \bmod n^2 \\
&= \frac{h^{a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j + r_i}\cdot\Theta\cdot(1+b\cdot n)}{h^{a\cdot\beta_i\cdot\sum_{j\in\mathbb{C}_{w_i}} r_j + r_i}} \bmod n^2 \\
&= \Theta\cdot(1+b\cdot n) \bmod n^2 \\
&= \prod_{j\in\mathbb{C}_{w_i}} (1+m_j\cdot n)^{a\cdot\beta_i}\cdot(1+b\cdot n) \bmod n^2.
\end{aligned} \tag{7.15}$$

Based on Section 7.4.2, m_j can be either 1 or 0 depending on if $\tau_j \subset \tau_{\mathtt{t}}$ or not. Therefore, we can write M_{w_i} as

$$\begin{aligned}
M_{w_i} &= (1+n)^{l_i\cdot a\cdot\beta_i}\cdot(1+b\cdot n) \\
&= (1+l_i\cdot a\cdot\beta_i\cdot n)\cdot(1+b\cdot n) \\
&= (1+b\cdot n+l_i\cdot a\cdot\beta_i\cdot n) \bmod n^2,
\end{aligned} \tag{7.16}$$

where l_i is the number of the cases that $m_j = 1$. Based on Definition 7.3.4, l_i is the number of overlapping hours between τ_{w_i} and $\tau_{\mathtt{t}}$, i.e., $l_i = \tau_{(\mathtt{t},w_i)}$. Moreover, $\mathcal{S}_2$ computes $\widetilde{M}_{w_i} = \mathcal{L}(M_{w_i}) = \frac{(M_{w_i}-1) \bmod n^2}{n}$, so we have

$$\widetilde{M}_{w_i} = a\cdot\tau_{(\mathtt{t},w_i)}\cdot\lfloor 10^5\cdot\ln(1-\mathtt{R}_{(\mathtt{t},w_i)})\rfloor + b. \tag{7.17}$$

We can see here, due to the existence of random numbers a and b, $\mathcal{S}_2$ cannot know $\tau_{(\mathtt{t},w_i)}\cdot\lfloor 10^5\cdot\ln(1-\mathtt{R}_{(\mathtt{t},w_i)})\rfloor$.

Finally, $\mathcal{S}_2$ receives $\{\widetilde{M}_{w_1}, \widetilde{M}_{w_2}, \ldots, \widetilde{M}_{w_d}\}$ from all the d workers in $\mathcal{W}$, and selects the top-k workers based on spatial-temporal similarity $\mathtt{Sim}_{(\mathtt{t},w_i)}$ for $k \geq 1$. Specifically, it is easy to know that $\widetilde{M}_{w_i}$ and $(1-\mathtt{R}_{(\mathtt{t},w_i)})^{\tau_{(\mathtt{t},w_i)}}$ have the same monotonicity. Therefore, the worker w_i who has a larger value of $\widetilde{M}_{w_i}$ will have a smaller value of $\mathtt{Sim}_{(\mathtt{t},w_i)}$ (i.e., $1-(1-\mathtt{R}_{(\mathtt{t},w_i)})^{\tau_{(\mathtt{t},w_i)}}$). Consequently, $\mathcal{S}_2$ can find the top-k workers ranked by $\widetilde{M}_{w_i}$ in an ascending order. And the k workers will be selected to perform the MCS task.

7.5 Security Analysis

In this section, we analyse the security properties of the proposed scheme. Particularly, based on the designed goals in Section 7.2.3, we will prove that the task's $\tau_{\mathtt{t}}$ and each worker w_i's τ_{w_i}, $\mathtt{Area}_{w_i}$ and $\mathtt{Sim}_{(\mathtt{t},w_i)}$ are privacy preserving in the whole process of worker selection.

• Privacy preservation of task's time duration $\tau_{\mathtt{t}}$: In the `TaskLun` phase, after receiving the overlapping time duration calculation table (i.e., $(c_1, c_2, \ldots, c_v)$), both $\mathcal{S}_1$ and w_i may try to infer if the plaintext m_j from c_j is 0 or 1 for $j \in [1, v]$. However, neither of them can link c_j with $m_j = 0$ or $m_j = 1$ according to the semantic security of modified Paillier PKE [10]. Besides, $\tau_{\mathtt{t}}$ does not appear in the `LolCal` and `SimComp` phases. Instead, worker w_i calculates $\tau_{(\mathtt{t},w_i)}$ based on $\tau_{\mathtt{t}}$ and encrypts $\tau_{(\mathtt{t},w_i)}$ in those two phases. Therefore, $\tau_{\mathtt{t}}$ is privacy-preserving in the whole process of worker selection.

• Privacy preservation of worker w_i's time duration τ_{w_i} and moving area $\mathtt{Area}_{w_i}$: In the proposed scheme, worker w_i does not use his/her time duration τ_{w_i} and moving area $\mathtt{Area}_{w_i}$ directly. Specifically, w_i will first check all the time slots to see if a particular time slot τ_j is included in τ_{w_i}. Next, w_i multiplies all the corresponding c_j(s) and computes $\widetilde{c}_{w_i} = \{\widetilde{C}_{(w_i,1)}, \widetilde{C}_{(w_i,2)}\}$ where $\widetilde{C}_{(w_i,1)} = C^{\beta_i}_{(w_i,1)}$ and $\widetilde{C}_{(w_i,2)} = C^{\beta_i}_{(w_i,2)}$. Besides, w_i will calculate the intersection area ratio $\mathtt{R}_{(\mathtt{t},w_i)}$ and then apply it in the

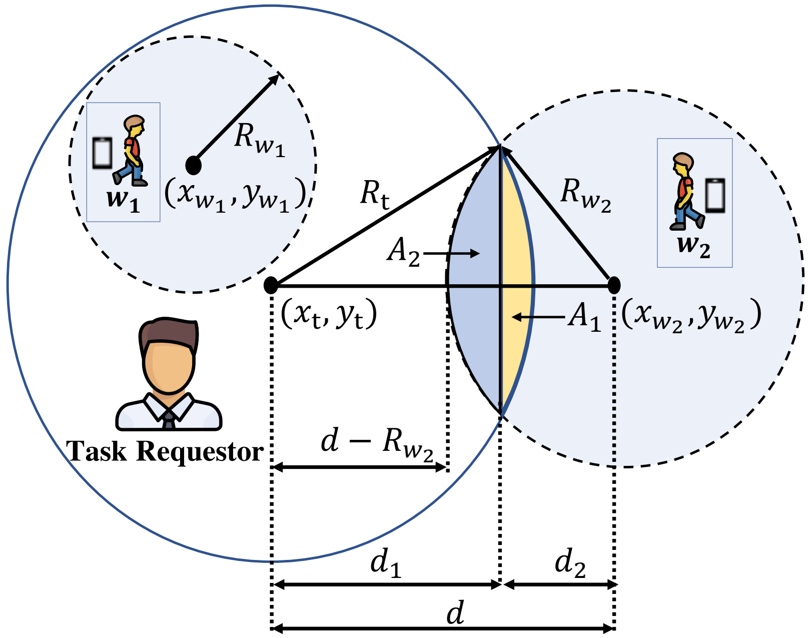


Figure 7.4: An illustration example of calculating the intersection area ratio $\mathtt{R}_{(\mathtt{t},w_i)}$ between the task and two workers w_1 and w_2.

calculation of $\widetilde{c}_{w_i}$. Based on the homomorphic properties of the modified Paillier PKE, $\widetilde{c}_{w_i}$ can still be considered as a valid ciphertext. So neither $\mathcal{S}_2$ nor $\mathcal{S}_2$ can discover useful information about $\tau_{\mathtt{t}}$ and $\mathtt{R}_{(\mathtt{t},w_i)}$, not even mention τ_{w_i} and $\mathtt{Area}_{w_i}$.

• Privacy preservation of the spatial-temporal similarity $\mathtt{Sim}_{(\mathtt{t},w_i)}$: In our scheme, $\mathcal{S}_1$ or $\mathcal{S}_2$ only keeps part of the secret key (i.e., s_1 or s_2). Therefore, given the assumption that the two cloud servers do not collude with each other, $\mathcal{S}_1$ can only partially decrypt the ciphertext and cannot discover any information about $\mathtt{Sim}_{(\mathtt{t},w_i)}$. In addition, $\mathcal{S}_1$ introduced two numbers a and b in `SimComp` phase. Therefore, $\mathcal{S}_2$ can only decrypt $\widetilde{M}_{w_i} = a \cdot \tau_{(\mathtt{t},w_i)} \cdot \lfloor 10^5 \cdot \ln(1 - \mathtt{R}_{(\mathtt{t},w_i)}) \rfloor + b$ and compare workers' spatial-temporal similarities. Given the fact that a and b are secretly kept by $\mathcal{S}_1$, $\mathcal{S}_2$ can never know the value of each worker's $\mathtt{Sim}_{(\mathtt{t},w_i)}$.

Overall, the task time duration $\tau_{\mathtt{t}}$ and worker w_i's spatial-temporal information (i.e., τ_{w_i}, $\mathtt{Area}_{w_i}$, and $\mathtt{Sim}_{(\mathtt{t},w_i)}$) are privacy-preserving and not disclosed to other entities in the proposed scheme.

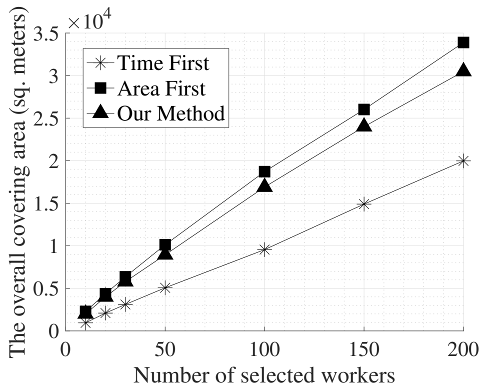
7.6 Performance Evaluation

In this section, we study the performance of the proposed worker selection scheme using a custom simulator built-in Python. Particularly, we simulate a MCS task $\mathcal{T} = (\texttt{Area}_{\texttt{t}}, \tau_{\texttt{t}})$, where we set $\texttt{Area}_{\texttt{t}} = 1000\ m^2$ and $\tau_{\texttt{t}} = 12$ hours. Moreover, we simulate 1000 heterogeneous workers with different spatial-temporal information. For each worker w_i where $i \in [1, 1000]$, we assume that the overlapping area $\texttt{Area}_{(\texttt{t},w_i)}$ follows the Gaussian distribution with 100 m^2 as the mean value and 50 m^2 as the standard deviation. In addition, the overlapping time duration $\tau_{(\texttt{t},w_i)}$ is set as a random number in the range of [1, 12]. Besides, in the experiment, we set $\kappa = 512$, thus $|p| = |q| = 512$ and $|n| = 1024$. We implement the simulator in Python and conduct the experiments on a Windows 10 machine learning with an Intel(R) Core(TM) i7-6700 CPU @3.6-GHz, and 16GB RAM. All the experiments are run 10000 times and the average results are reported.

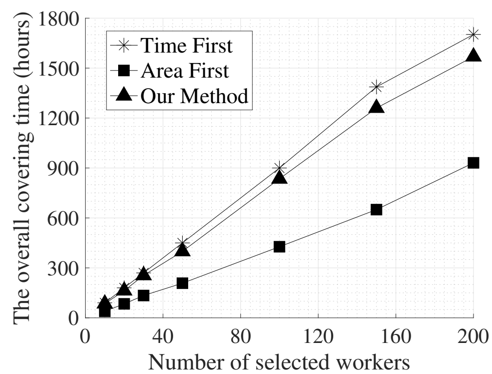
To evaluate the proposed scheme, we also implement two other approaches as the baselines for comparison purposes. They are (1) Time First approach and (2) Area First approach. Specifically, in the Time First approach, the MCS platform first ranks the workers based on the overlapping time duration $\tau_{(\texttt{t},w_i)}$ and selects the workers with top $\tau_{(\texttt{t},w_i)}$ values. On the other hand, in the Area First approach, the platform selects workers only based on the intersection area $\texttt{Area}_{(\texttt{t},w_i)}$ between the task and the workers.

Let $\mathcal{W}_{\texttt{sel}} = \{w_1, w_2, ..., w_k\}$ denote the group of k workers selected by different approaches. Essentially, $\mathcal{W}$ is the group of top-k workers either by $\texttt{Area}_{(\texttt{t},w_i)}$ (i.e., in Area First approach), $\tau_{(\texttt{t},w_i)}$ (i.e., in Time First approach), or $\texttt{Sim}_{(\texttt{t},w_i)}$ (i.e., in our proposed scheme).

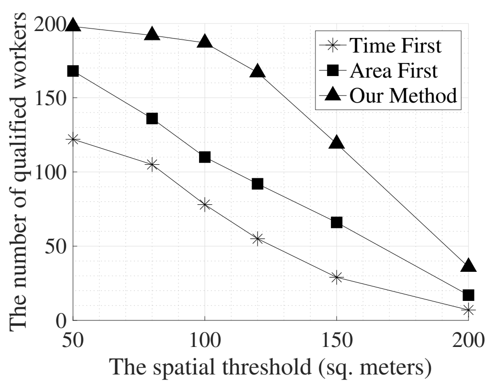
In the experiment, first, we define two simple evaluation metrics, namely the overall



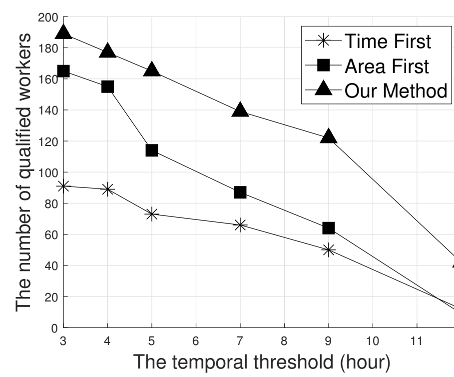
(a) The overall spatial coverage $\texttt{Area}_{\texttt{all}}$ with the number of workers varies from 10 to 200



(b) The overall temporal coverage $\tau_{\texttt{all}}$ with the number of workers varies from 10 to 200



(c) The number of qualified workers with $\tau_{\texttt{bound}} = 6$ hours and $\texttt{Area}_{\texttt{bound}}$ varies from 50 to 200 m^2.



(d) The number of qualified workers with $\texttt{Area}_{\texttt{bound}} = 100$ m^2 hours and $\tau_{\texttt{bound}}$ varies from 2 to 12 hours.

Figure 7.5: The experimental results for comparing the performance of the proposed worker selection scheme with two baseline approaches.

spatial coverage $\texttt{Area}_{\texttt{all}}$ and the overall temporal coverage $\tau_{\texttt{all}}$, where

$$\begin{cases} \texttt{Area}_{\texttt{all}} = \sum_{i=1}^{u} \texttt{Area}_{(\texttt{t},w_i)}, \\ \tau_{\texttt{all}} = \sum_{i=1}^{u} \tau_{(\texttt{t},w_i)}. \end{cases} \tag{7.18}$$

Here, we know that $\texttt{Area}_{\texttt{all}}$ and $\tau_{\texttt{all}}$ indicate the overall spatial- and temporal-coverage for the task by all the selected workers. From Fig. 7.5 (a) and (b), we can see that, both the Time First and the Area First approaches only consider one dimension in selecting the qualified workers. Therefore, although the Time First approach can occupy the longest time duration for the sensing task, it only covers the minimal spatial area. Similarly, the Area First approach can cover the maximal sensing area, but it shows the worst performance in temporal coverage. The low spatial- or temporal-coverage may lead to poor data quality, such as incomplete data, inaccurate data, or useless data. However, our proposed scheme considers spatial and temporal information simultaneously. It can achieve high spatial-coverage as that in the Area First approach and high temporal-coverage as that in the Time First approach, which guarantees the high quality of the sensing data.

Next, we set a spatial lower boundary $\texttt{Area}_{\texttt{boud}}$ and a temporal lower boundary $\tau_{\texttt{bound}}$. We select the top-200 workers (i.e., $\mathcal{W}_{\texttt{sel}} = \{w_1, w_2, ..., w_{200}\}$) based on three different approaches. We say a worker w_i satisfies the two boundaries if $\texttt{Area}_{(\texttt{t},w_i)} \geq \texttt{Area}_{\texttt{boud}}$ and $\tau_{(\texttt{t},w_i)} \geq \tau_{\texttt{bound}}$. Then, we check how many workers in $\mathcal{W}_{\texttt{sel}}$ satisfy both the spatial lower boundary $\texttt{Area}_{\texttt{boud}}$ and the temporal lower boundary $\tau_{\texttt{bound}}$. In Fig. 7.5 (c), we set $\tau_{\texttt{bound}} = 6$ hours and $\texttt{Area}_{\texttt{boud}}$ varies from 50 to 200 m^2. In Fig. 7.5 (d), we set $\texttt{Area}_{\texttt{boud}} = 100$ m^2 and $\tau_{\texttt{bound}}$ varies from 2 to 12 hours. From both Fig. 7.5 (c) and (d), we can see that our proposed scheme outperforms the other two baselines. Particularly, our scheme can select much more qualified workers under different spatial and temporal boundaries. Given the same task budget, our approach can select more reliable workers and provide more trustable sensing data.

7.7 Related Work

Zhu et al. [148] studied the problem of mobile vehicle selection in MCS to collect maximum data in urban environment based on future mobility. They first formulated the recruitment problem as a maximum data limited budget problem. Then, a deep learning-based framework based on mobile vehicles is proposed to collect sensing data in real-world application scenarios. The deep learning algorithm can predict vehicle mobility for a future time period in an offline manner. Next, they designed a greedy online algorithm to select vehicle with a limited budget for the NP-complete problem. Extensive evaluations are conducted on real-world datasets to validate the efficiency of the proposed solution. Hu et al. [42] studied a combinatorial pinning zero-determinant strategy for achieving the cost-efficient worker selection in MCS. By considering the spatial diversity and temporal dynamics of MCS participating workers, their proposed method can minimize the total expected utilities of the workers throughout all sensing regions for each time interval, without affecting workers' strategies. In addition, the designed method extends the application scenarios of the classical zero-determinant strategy from two-player simultaneous-move games to multiple-heterogeneous-player sequential move ones. Yu et al. [125] studied the problem of cost-minimizing task allocation problem with spatial-temporal factors in MCS. In specific, the MCS platform provides a set of tasks with different locations and time slots. The platform plans to select the optimal workers based on the spatial-temporal information to achieve the minimum cost. The authors proved that it is a NP-hardness problem. Next, a classical primal-dual algorithm was proposed by a heuristic algorithm. Extensive simulations are deployed to verify the effectiveness of the proposed algorithm. Kong et al. [51] exploited the worker selection problem by designing a spatial-temporal incentive mechanism for smartphone users. They first utilize the logistic regression algorithm to mine user preferences from historical information. Next, a sub-module approximation approach was developed to solve the worker selection problem. Moreover, the authors designed a winner auction algorithm based on the reverse auction mechanism to avoid the critical payment allocation issue. The performance of the proposed algorithms are evaluated by both theoretical analyses and extensive simulations.

7.8 Concluding Remarks

In this chapter, we have proposed a privacy-preserving top-k worker selection scheme in terms of spatial-temporal similarity in MCS services. First, we introduced the definition and calculation of spatial-temporal similarity between participating workers and the MCS task in terms of intersection area ratio and overlapping time duration. Next, we designed a privacy-preserving approach for protecting participants' sensitive information based on the modified Paillier encryption technique. Particularly, the workers do not need to directly provide their moving area and the corresponding time duration in the moving area, where the MCS platform can still select qualified workers for the task. In our approach, the task requestor's temporal information and the workers' spatial-temporal information are protected and will not be disclosed to other parties. Finally, we validated the reliability of our proposed scheme by experimental results with different scenarios. For future work, we will study the problem of achieving the maximal spatial/temporal diversities in MCS applications.

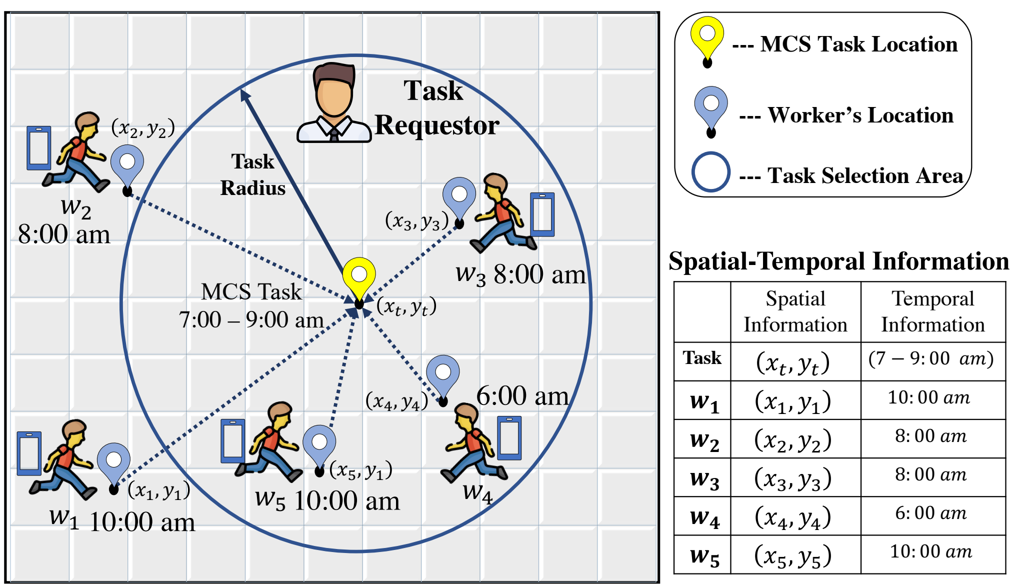
Chapter 8

Privacy-preserving Worker Selection in MCS over Spatial-temporal Constraints

8.1 Introduction

In the previous chapter, we proposed a privacy-preserving worker selection scheme regarding the spatial-temporal similarity between workers and the MCS task. Aiming at extending the research scope of the previous chapter, in this chapter, we study privacy-preserving worker selection problem in terms of the spatial-temporal constraint. Typically, each MCS task is requested at a given location and is constrained by a certain time interval. The workers need to physically move to a particular location during the valid time interval to complete the task. Therefore, the MCS platform should select suitable workers for each task from the perspective of both spatial and temporal constraints.

• **Spatial constraint**. For the assignment of each work, the task requestor needs to provide some incentives to the selected worker after he/she successfully completes the task. The incentives are usually monetary or in-kind rewards that can engage and motivate workers for providing timely and high-quality services. There are numerous existing studies focusing on identifying how many incentives should be given to each worker. A commonly-agreed approach is that the incentive should at least cover workers' costs during the conduction of the MCS task. The costs include cognitive resources, device battery, and the traveling costs [47]. In this work, similar to [112, 91], we assume that the incentive provided for each worker is proportional to his/her traveling distance. In addition, for each MCS task, the requestor's budgets are limited. The task requestor wants to select workers that are closer to the sensing location, so that they require lower incentives. The distance workers may not be considered. Therefore in this work, each task requestor needs to predefine a maximum distance (i.e., a maximum price of the incentive) where only the workers within this area can be selected.



	Spatial Information	Temporal Information
Task	(x_t, y_t)	$(7-9{:}00\ am)$
w_1	(x_1, y_1)	$10{:}00\ am$
w_2	(x_2, y_2)	$8{:}00\ am$
w_3	(x_3, y_3)	$8{:}00\ am$
w_4	(x_4, y_4)	$6{:}00\ am$
w_5	(x_5, y_5)	$10{:}00\ am$

Figure 8.1: An example of spatial-temporal constraint based worker selection over a sequence of activities.

• **Temporal constraint**. Usually, an MCS task is highly time-related. For example, a task requestor wants to check if there is any available parking spot from 12:00 pm to 1:00 pm in a busy shopping district. A driver wants to evaluate the traffic congestion situation from 4:30 pm to 5:00 pm on a crowded street. Real-time sensing data should be collected specifically during the given time period, otherwise, the sensed information may be useless for the task requestors. Therefore, it is essential to make sure that the time when the worker can arrive at the sensing location falls within the range of the task's time duration. On the other hand, in [16], the researchers observed that the workers dynamically move towards some directions, and they only prefer to accept the tasks along their scheduled path. In other words, the workers do not want to wait for a long time to perform the task, which will certainly affect their original plan. Therefore, the MCS platform should take the temporal factors into consideration as well and should assign MCS tasks to the most suitable workers based on the spatial-temporal constraints.

A motivating example is shown in Fig. 8.1 to demonstrate how spatial and temporal constraints are being considered in a real-world MCS service.

In Fig. 8.1, there is an MCS task $\mathcal{T}$ initialized by a task requestor and five MCS workers $\{w_1, \ldots, w_5\}$. The spatial-temporal information for both $\mathcal{T}$ and the workers are shown in the table. Task $\mathcal{T}$ is associated with a specific location (x_t, y_t), a valid time duration (t_s, t_e), and a maximum covering distance $R_{\mathtt{t}}$, e.g., $\mathcal{T}_1$ is to monitor the traffic situation on a busy street from 7:00 am to 9:00 am. Each worker reports a location and the corresponding time arrived at the location in advance. For example, worker w_1 will arrive at location (x_1, y_1) at $t_1 =$ 10:00 am. The MCS platform needs to allocate the task to the workers who meet both the spatial constraint and the temporal constraint. In particular, from the figure, it is easy to tell that workers w_3, w_4, and w_5 meet the spatial constraint since their distances to the sensing location are smaller than the maximum sensing coverage $R_{\mathtt{t}}$. Furthermore, let us assume that it will take 10 mins, 5 mins, and 10 mins for workers w_3, w_4, w_5 to travel to the task location, respectively. Then, the arrival times for them are 8:10 am, 6:05 am, and 10:10 am. Consequently, we know that only w_3 meets the temporal constraint of the sensing task since her arrival time 8:10 am falls within the range of the task's duration (i.e., 7:00 am to 9:00 am). For worker w_4, her arrival time is 6:05 am, which means that w_4 needs to wait for nearly 1 hour before the task starts. The long waiting time stops w_4 from participating in the sensing task. On the other hand, w_5's arrival time is 10:10 am, even 1 hour after the task is finished. Therefore, w_5 is not a suitable candidate either. In conclusion, worker w_3 is the only participant who meets both the spatial and temporal constraints for the sensing task.

Rather than the spatial-temporal factors, another major issue that prevents the workers from taking part in the sensing tasks is the potential privacy disclosure. As we see in the previous example, both the task requestor and the workers need to submit their accurate spatial-temporal data, which contains sensitive personal information and can be used by attackers to infer a person's daily routine and life activities [140]. Essentially, the MCS platform should protect workers' sensitive spatial-temporal information while still selecting suitable workers based on the spatial-temporal constraint.

To address the above problem, we propose a novel privacy-preserving scheme for worker selection based on the spatial-temporal constraints. The overall contributions of this work are three-fold. First, we formally propose and define a practical worker selection scenario in MCS services where workers should meet both the spatial and temporal constraints. Second, we design a symmetric homomorphic encryption (SHE) [144, 68] based privacy-preserving worker selection scheme, which can protect participants' sensitive information from being disclosed. Third, we conduct extensive experiments to validate the effectiveness of the proposed scheme.

The roadmap of the rest of this chapter is organized as follows. In Section 8.2, we describe our system model, security model, and design goals. In Section 8.3, we recall the symmetric homomorphic encryption approach and demonstrate the proposed scheme in detail. In Section 8.4 and 8.5, we discuss the security analysis and evaluate the performance of the designed scheme, respectively. In Section 8.6, we introduce some related work. In Section 8.7, we recap the conclusions and propose some potential research directions as future work.

8.2 Models and Design Goals

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

8.2.1 System Model

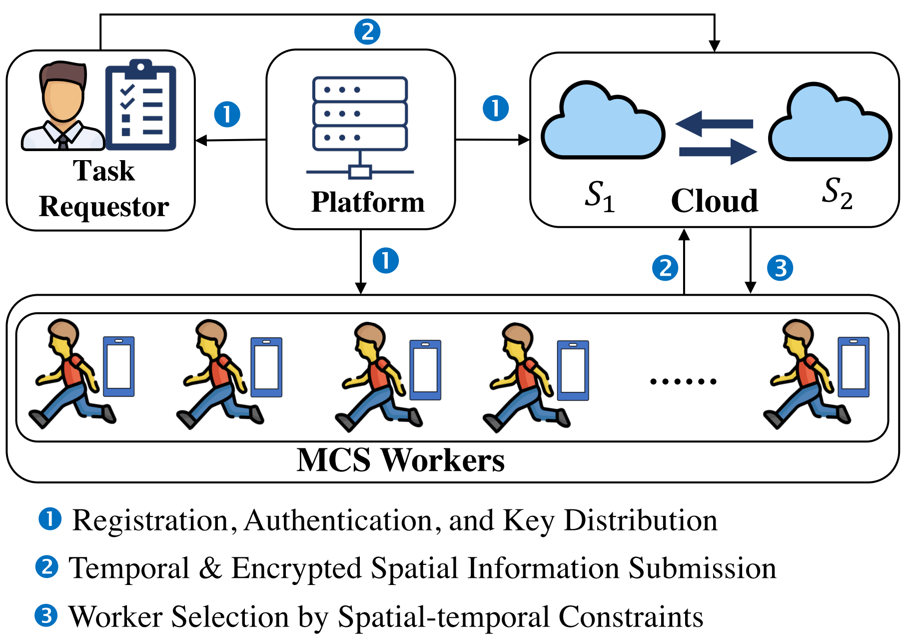


Figure 8.2: The system model under consideration in chapter 8.

As shown in Fig. 8.2, the system model in this work mainly consists of four entities, namely, a MCS platform $\mathcal{P}$, a task requestor $\mathcal{R}$, a cloud platform with two servers $\{\mathcal{S}_1, \mathcal{S}_2\}$, and a group of participating workers $\mathcal{W} = \{w_1, w_2, \dots\}$. Primarily, there are steps in the overall workflow of the proposed worker selection scheme. The MCS platform first authenticates the identities of both the workers and the task requestors, and then registers them in the system. Next, the task requestors and the workers need to send the encrypted sensitive information to the cloud servers. Finally, based on spatial-temporal constraints, the cloud servers can select suitable workers for the task. The details of each entity are described as follows.

• The MCS Platform $\mathcal{P}$: $\mathcal{P}$ is the service provider who is responsible for bootstrapping the entire MCS service. Essentially, $\mathcal{P}$ registers both the task requestor and workers before the sensing tasks start. Besides, $\mathcal{P}$ generates and distributes keys to different entities, so that a certain task can be completed cooperatively. After that, $\mathcal{P}$ just stays offline.

• Cloud Servers $\{\mathcal{S}_1, \mathcal{S}_2\}$: There are two cloud servers $\{S_1, S_2\}$ in our system model. Basically, S_1 and S_2 work together to select qualified workers who meet both the spatial and temporal constraints as we mentioned earlier. After worker selection, $\{\mathcal{S}_1, \mathcal{S}_2\}$ collect the sensing data from the workers, and then send the data to the task requestor.

• Task Requestor $\mathcal{R}$: In MCS services, the task requestor $\mathcal{R}$ is the task initiator (e.g., government, companies, or individuals). Before each task begins, $\mathcal{R}$ needs to define three attributes, $L_{\mathtt{t}}, \tau_{\mathtt{t}}$, and $R_{\mathtt{t}}$. Here, $L_{\mathtt{t}} = (x_{\mathtt{t}}, y_{\mathtt{t}})$ is the task location in the 2-dimensional space; $\tau_{\mathtt{t}} = (t_s, t_e)$ is the task's time duration where t_s and t_e are the start time and end time for the task, respectively; $R_{\mathtt{t}}$ is the radius of the maximum covering area for the task. Next, $\mathcal{R}$ needs to encrypt sensitive spatial-temporal information and then sends the encrypted data to the cloud servers for worker selection.

• Workers $\mathcal{W}$: $\mathcal{W} = \{w_1, w_2, \dots\}$ are the participating workers who wish to perform the MCS tasks and earn some rewards. Each worker w_i needs to submit his/her location $L_{w_i} = (x_i, y_i)$ and the corresponding time t_{w_i} when w_i will arrive L_{w_i}. Similarly, each worker w_i will encrypt the sensitive personal information and then send the encrypted information to the cloud servers. In this work, only the workers who meet both the spatial and temporal constraints will be selected by the MCS platform for conducting the task. After being selected, each worker will physically travel to the task location and start collecting real-time sensing data.

8.2.2 Security Model

In our security model, we consider $\mathcal{P}$, $\mathcal{W}$ and $\mathcal{R}$ to be trustable. However, we consider the two cloud servers $\{\mathcal{S}_1, \mathcal{S}_2\}$ to be semi-honest. In other words, $\mathcal{S}_1$ and $\mathcal{S}_2$ will strictly follow the protocol, yet may be curious to learn $\mathcal{R}$ or $\mathcal{W}$'s spatial-temporal information. For $\mathcal{W}$, we assume that they are selfish to maximize their own incentives. Hence, if a worker provides false information, he/she may have a higher chance of being selected. However, similar to [136], we assume that the reputation-based evaluation system is used to update workers' trustabilities. Only the workers with high trustabilities can be considered by the system. Therefore, from a long-term perspective, each worker should honestly submit their personal spatial-temporal information as accurately as possible. Besides, there is no collusion between $\mathcal{S}_1$ and $\mathcal{S}_2$. In addition, neither the task requestor $\mathcal{R}$ nor any worker can collude with the two cloud servers.

It is worth noting that outside attackers may exploit other vulnerabilities of the MCS system, as this work focuses on privacy preservation, those attacks are beyond the scope of this thesis and will be discussed in our future work.

8.2.3 Design Goals

Overall, the goal of this work is to securely and efficiently select workers for an MCS task who meet both the spatial and temporal constraints. Specifically, the following objectives should be satisfied.

• **Privacy**: In the system model, the task requestor $\mathcal{R}$ and the workers $\mathcal{W}$ are required to submit both their spatial information and temporal information to the cloud servers. In our security model, we assume that the cloud servers cannot track a person's daily life and infer his/her personal routine as long as the temporal and the spatial information are not leaked at the same time. More specifically, for simplicity purposes, we consider that both $\mathcal{R}$ and $\mathcal{W}$'s spatial information (i.e., $L_{\mathtt{t}}$ and L_{w_i}) are sensitive and should not be disclosed to the cloud servers. On the other hand, their temporal information (i.e., $\tau_{\mathtt{t}}$ and t_{w_i}) are not private and can be submitted in their plaintext. The rationale behind this assumption is that in a real-world MCS scenario, a participant's temporal information is more likely to be random. Therefore, an attacker cannot get any useful information by only knowing the time. However, the spatial information is often fixed and contains more contextual insights about a person's behaviour. For example, by knowing a person's exact location (e.g., office, shopping district, hospital, home, school, gym, etc.), an attacker can easily guess the victim's occupation, habit, daily behaviour, educational background, health condition, and incoming even without any knowledge about the corresponding temporal information [140]. In a nutshell, in our security model, we consider the spatial information as sensitive and should be protected. The temporal information is non-sensitive and can be leaked by other entities.

- **Spatial-temporal Constraint based Worker Selection**: Here, we formally define the spatial constraint and temporal constraint for worker selection.

Definition 8.2.1. *(**The Spatial Constraint**): An MCS task can be denoted as $\mathcal{T} = \langle L_{\mathtt{t}}, \tau_{\mathtt{t}}, R_{\mathtt{t}} \rangle$, where $L_{\mathtt{t}} = (x_{\mathtt{t}}, y_{\mathtt{t}})$ is the task location, $\tau_{\mathtt{t}} = (t_s, t_e)$ is the task's time duration, and $R_{\mathtt{t}}$ is the radius of the maximum covering area. For a given worker $w_i = \langle L_{w_i}, t_{w_i}, v_{w_i} \rangle$ where L_{w_i} is the worker's location, t_{w_i} is the time when w_i will be in L_{w_i}, and v_{w_i} is w_i's travel speed. We say that w_i meets the spatial constraint with $\mathcal{T}$ (denoted by $w_i \cap_s \mathcal{T} \neq \emptyset$) if*

$$\boldsymbol{dist}(L_{\mathtt{t}}, L_{w_i}) = \sqrt{(x_{\mathtt{t}} - x_i)^2 + (y_{\mathtt{t}} - y_i)^2} \leq R_{\mathtt{t}}. \tag{8.1}$$

Definition 8.2.2. ***(The Temporal Constraint)**: Given an MCS task $\mathcal{T} = \langle L_t, \tau_t, R_t \rangle$ and a worker $w_i = \langle L_{w_i}, t_{w_i}, v_{w_i} \rangle$, first we can calculate the time $t_{(w_i,\mathcal{T})}$ when w_i can arrive at the task locatin L_t as*

$$t_{(w_i,\mathcal{T})} = t_{w_i} + \frac{\boldsymbol{dist}(L_t, L_{w_i})}{v_{w_i}}. \tag{8.2}$$

Next, we say w_i meets the temporal constraint with $\mathcal{T}$ (denoted by $w_i \cap_t \mathcal{T} \neq \emptyset$) if $t_s \leq t_{(w_i,\mathcal{T})} \leq t_e$, then it is easy to get

$$v_{w_i} \cdot (t_s - t_{w_i}) \leq \boldsymbol{dist}(L_t, L_{w_i}) \leq v_{w_i} \cdot (t_e - t_{w_i}). \tag{8.3}$$

Definition 8.2.3. ***(Worker Selection over Spatial-temporal Constraints)**: Given an MCS task $\mathcal{T} = \langle L_t, \tau_t, R_t \rangle$ and a large group of workers $\mathcal{W} = \{w_1, w_2, \dots\}$, the problem of spatial-temporal constraint based worker selection is to assign $\mathcal{T}$ with a set of workers $\mathcal{W}_s \subseteq \mathcal{W}$, such that for $\forall w_i \in \mathcal{W}_s$, w_i meets both the spatial constraint and temporal constraint, i.e., $w_i \cap_s \mathcal{T} \neq \emptyset$ and $w_i \cap_t \mathcal{T} \neq \emptyset$. Based on Eq. (8.2) and Eq. (8.3), we know worker w_i meets both the spatial and temporal constraints with the sensing task iff*

$$\Delta_1 \leq \boldsymbol{dist}(L_t, L_{w_i}) \leq \min{(R_t, \Delta_2)}, \tag{8.4}$$

where $\Delta_1 = v_{w_i} \cdot (t_s - t_{w_i})$ and $\Delta_2 = v_{w_i} \cdot (t_e - t_{w_i})$.

• **Efficiency**: In the real-world MCS services, both the workers and the task arrive at/leave the system in a streaming manner. Due to the highly dynamic nature of MCS participants, the decision of worker assignment should be made as fast as possible. In addition, to preserve participants' privacy, certain encryption approaches should be applied, which will bring extra computation and communication costs. Therefore, our proposed scheme should include some effective filtering algorithms to enable efficient worker selection in a real-time fashion.

8.3 The Proposed Scheme

In this section, we describe the proposed scheme based on the symmetric homomorphic encryption (SHE).

8.3.1 Privacy-preserving Worker Selection

Given the definitions of spatial and temporal constraints in Section 8.2.3, we design our privacy-preserving worker selection scheme as follows.

8.3.1.1 System Initialization

At first, the MCS platform $\mathcal{P}$ registers all the task requestors and workers by verifying their identities. Next, $\mathcal{P}$ runs the key generation algorithm $\texttt{KeyGen}(k_0, k_1, k_2)$ to create the public parameters $pk = (k_0, k_1, k_2, \mathcal{N})$ and the secret key $sk = (p, \mathcal{L})$. Next, $\mathcal{P}$ sends the secret key sk to the task requestor, workers, and the second cloud $\mathcal{S}_2$. After initializing the entire system, $\mathcal{P}$ just stays offline and will not participate in the real-time MCS services.

8.3.1.2 Encrypted Information Submission

In this phase, both the task requstor $\mathcal{R}$ and the workers need to submit the encrypted information to the cloud servers for worker selection. The details of this phase are discussed as below.

• For the Task Requestor $\mathcal{R}$: Given an MCS task $\mathcal{T} = \langle L_{\mathtt{t}}, \tau_{\mathtt{t}}, R_{\mathtt{t}} \rangle$ where $L_{\mathtt{t}} = (x_{\mathtt{t}}, y_{\mathtt{t}})$ and $\tau_{\mathtt{t}} = (t_s, t_e)$, $\mathcal{R}$ needs to protect the spatial information (i.e., $x_{\mathtt{t}}$ and $y_{\mathtt{t}}$) from being disclosed, while the temporal information (i.e., t_s and t_e) can be transmitted in plaintexts. Specifically, using the secret key sk generated by $\mathcal{P}$, $\mathcal{R}$ prepares $C_{\mathtt{t}}$ as follows:

$$C_{\mathtt{t}} = \begin{cases} c_1 = \mathtt{E}(x_{\mathtt{t}}^2 + y_{\mathtt{t}}^2) \\ c_2 = \mathtt{E}(x_{\mathtt{t}}) \\ c_3 = \mathtt{E}(y_{\mathtt{t}}). \end{cases} \tag{8.5}$$

After that, $\mathtt{R}$ sends $(C_{\mathtt{t}}, \tau_{\mathtt{t}}, R_{\mathtt{t}})$ to the cloud server $\mathcal{S}_1$.

• For the Worker w_i: Similarly, for each worker $w_i = \langle L_{w_i}, t_{w_i}, v_{w_i} \rangle$, he/she only needs to encrypt the spatial information (i.e., L_{w_i}) whereas both t_{w_i} and v_{w_i} can be disclosed to the cloud servers. In particular, given the secret key sk generated by $\mathcal{P}$, w_i calculates C_{w_i} as follows:

$$C_{w_i} = \begin{cases} c_4 = \mathtt{E}(x_i^2 + y_i^2) \\ c_5 = \mathtt{E}(-2x_i) \\ c_6 = \mathtt{E}(-2y_i). \end{cases} \tag{8.6}$$

Next, w_i distributes $(C_{w_i}, t_{w_i}, v_{w_i})$ to the cloud server $\mathcal{S}_1$.

8.3.1.3 Spatial-temporal filtering for Candidate Selection

After receiving all the information from both the task requestor $\mathcal{R}$ and the workers, the cloud server $\mathcal{S}_1$ can perform some pre-filtering processes to filter out the unqualified workers based on the submitted plaintexts. Consequently, $\mathcal{S}_1$ can accelerate the overall speed of the worker selection procedure, which is extremely significant in real-world MCS services.

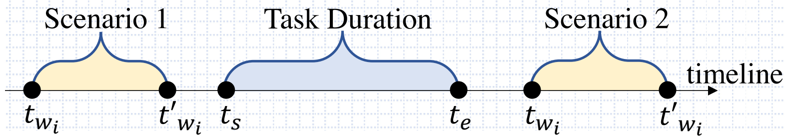


Figure 8.3: The illustration of the two filtering scenarios

In particular, given a task $\mathcal{T}$, a worker w_i and their submitted plaintexts, $\mathcal{S}_1$ can check the following two scenarios:

$$\begin{cases} \text{Scenario 1:} \;\; R_{\mathtt{t}} \leq v_{w_i} \cdot (t_s - t_{w_i}) = \Delta_1 \\ \text{Scenario 2:} \;\; t_{w_i} \geq t_e. \end{cases} \tag{8.7}$$

If w_i meets any of the above two scenarios, then w_i can be excluded from being considered as the potential candidate. Otherwise, $\mathcal{S}_1$ needs to follow the steps in Section 8.3 to select the workers who meet both the spatial-temporal constraints. Next, we prove the correctness of the filtering process as follows.

Proof: Given Eq. (8.2), we know that worker's arrival time $t_{(w_i,\mathcal{T})} = t_{w_i} + \frac{\mathtt{dist}(L_{\mathtt{t}}, L_{w_i})}{v_{w_i}}$. Therefore, let $t'_{w_i} = t_{w_i} + \frac{R_{\mathtt{t}}}{v_{w_i}}$, if w_i meets the spatial constraint (i.e., $\mathtt{dist}(L_{\mathtt{t}}, L_{w_i}) \leq R_{\mathtt{t}}$), it is easy to get $t_{w_i} \leq t_{(w_i,\mathcal{T})} \leq t'_{w_i}$. On the other hand, we know if w_i meets the temporal constraint, then based on Eq. (8.3), we have $v_{w_i} \cdot (t_s - t_{w_i}) \leq \mathtt{dist}(L_{\mathtt{t}}, L_{w_i}) \leq v_{w_i} \cdot (t_e - t_{w_i})$. Therefore, it is easy to get $t_s \leq t_{w_i} + \frac{\mathtt{dist}(L_{\mathtt{t}}, L_{w_i})}{v_{w_i}} \leq t_e$, which means $t_s \leq t_{(w_i,\mathcal{T})} \leq t_e$. In other words, the time duration $[t_{w_i}, t'_{w_i}]$ should overlap with the task time duration $[t_s, t_e]$. There are two scenarios that w_i does not meet this condition, which are shown in Fig. 8.3. In conclusion, if w_i meet any of the two scenarios in Eq. (8.7), he/she will not be selected as the candidate for performing the task due to w_i does not meet either the spatial/temporal constraint or both constraints.

8.3.1.4 Spatial-temporal Constraints Evaluation

After filtering out all the unqualified workers, $\mathcal{S}_1$ needs to find the workers from the rest of the candidates who meet both the spatial and temporal constraints.

At first, based on the *Homomorphic Addition-I* and *Homomorphic Multiplication-II* in Section 3.2.4, $\mathcal{S}_1$ calculates $\Delta_3 = \mathtt{E}(\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i}))$ as follows:

$$\begin{aligned}\Delta_3 &= c_1 + c_4 + c_2 \cdot c_5 + c_3 \cdot c_6 \\ &= \mathtt{E}(x_{\mathtt{t}}^2 + y_{\mathtt{t}}^2) + \mathtt{E}(x_i^2 + y_i^2) \\ &\quad + \mathtt{E}(-2x_i) \cdot \mathtt{E}(x_{\mathtt{t}}) + \mathtt{E}(-2y_i) \cdot \mathtt{E}(y_{\mathtt{t}}) \\ &= \mathtt{E}(x_{\mathtt{t}}^2 + y_{\mathtt{t}}^2 + x_i^2 + y_i^2 - 2x_i \cdot x_{\mathtt{t}} - 2y_i \cdot y_{\mathtt{t}}) \\ &= \mathtt{E}(\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})).\end{aligned} \tag{8.8}$$

Next, $\mathcal{S}_1$ generates a random number $r_0 \in (0, 2^{k_1-1})$, calculates $\Delta_1 = v_{w_i} \cdot (t_s - t_{w_i})$ and $\Delta_2 = v_{w_i} \cdot (t_e - t_{w_i})$, and compares $R_{\mathtt{t}}$ with Δ_2. If $\Delta_2 \geq R_{\mathtt{t}}$, then based on the *Homomorphic Addition-I*, *Homomorphic Multiplication-I* and *II* in Section 3.2.4, $\mathcal{S}_1$ computes Δ_4 as follows.

$$\begin{aligned}\Delta_4 &= r_0 \cdot (\Delta_3 + (-R_{\mathtt{t}}^2))(\Delta_3 + (-\Delta_1^2)) \\ &= r_0(\mathtt{E}(\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})) - R_{\mathtt{t}}^2)(\mathtt{E}(\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})) - \Delta_1^2) \\ &= \mathtt{E}(r_0(\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})) - R_{\mathtt{t}}^2)(\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})) - \Delta_1^2)).\end{aligned} \tag{8.9}$$

Otherwise, if $\Delta_2 < R_{\mathtt{t}}$, then $\mathcal{S}_1$ calculates Δ_4 as below.

$$\begin{aligned}\Delta_4 &= r_0 \cdot (\Delta_3 + (-\Delta_2^2))(\Delta_3 + (-\Delta_1^2)) \\ &= r_0(\mathtt{E}(\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})) - \Delta_2^2)(\mathtt{E}(\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})) - \Delta_1^2) \\ &= \mathtt{E}(r_0(\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})) - \Delta_2^2)(\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})) - \Delta_1^2)).\end{aligned} \tag{8.10}$$

After that, $\mathcal{S}_1$ sends Δ_4 to $\mathcal{S}_2$. Upon receiving Δ_4, $\mathcal{S}_2$ can recover $\Delta_5 = \mathtt{D}(\Delta_4)$ with the secret key $sk = (p, q, \mathcal{L})$ based on the SHE decryption algorithm in Section 3.2.4. Based on the scheme described above, we know that

$$\Delta_5 = \begin{cases} r_0 \cdot (\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i}) - R_{\mathtt{t}}^2) \cdot (\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i}) - \Delta_1^2), \text{if } R_{\mathtt{t}} < \Delta_2. \\ r_0 \cdot (\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i}) - \Delta_2^2) \cdot (\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i}) - \Delta_1^2), \text{otherwise}. \end{cases} \tag{8.11}$$

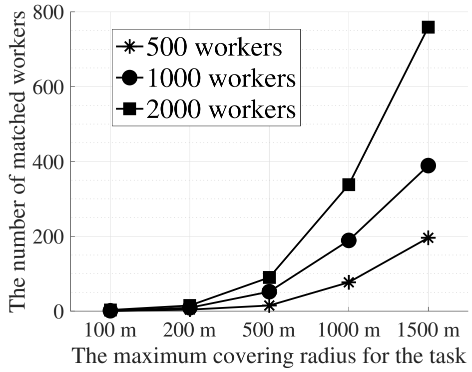
If $\Delta_5 < 0$, then w_i meets the spatial-temporal constraint of the task and can be selected to perform the task. Otherwise, w_i does not meet the constraint and will not be selected. As we know $\Delta_1 = v_{w_i} \cdot (t_s - t_{w_i})$ and $\Delta_2 = v_{w_i} \cdot (t_e - t_{w_i})$, then we can get $\Delta_2 > \Delta_1$ due to $t_e > t_s$. Besides, we know that after the pre-filtering process, for all the workers we get $R_{\mathtt{t}} > \Delta_1$. Given the fact that $r_0 > 0$, if we get $\Delta_5 < 0$, then it is easy to have $\Delta_1 < \mathtt{dist}(L_{\mathtt{t}}, L_{w_i}) < R_{\mathtt{t}}$ or $\Delta_1 < \mathtt{dist}(L_{\mathtt{t}}, L_{w_i}) < \Delta_2$, which proves that w_i meets the spatial-temporal constraints.

8.4 Security Analysis

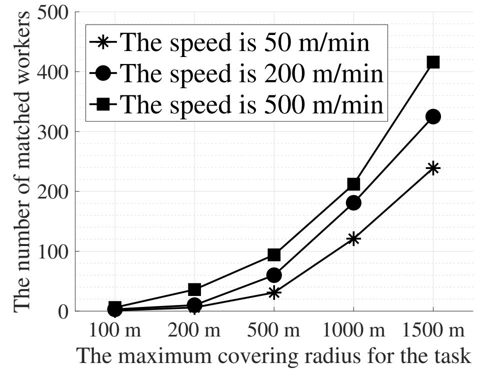
In this section, we analyze the security properties of the proposed scheme. In particular, based on the designed goals in Section 8.2.3, we prove that for both workers and the task requestor, their spatial information are protected from being disclosed in worker selection.

In the phase *Encrypted Information Submission*, after receiving the encrypted information $C_{\mathtt{t}}$ and C_{w_i} from the task requestor $\mathcal{R}$ and each worker w_i, the cloud server $\mathcal{S}_1$ may try to infer the original spatial information $L_{\mathtt{t}} = (x_{\mathtt{t}}, y_{\mathtt{t}})$ and $L_{w_i} = (x_i, y_i)$. However, SHE is a symmetric homomorphic encryption, which is proved to be indistinguishable chosen plaintext attack (IND-CPA) secure [68, 144]. Therefore, without the secret key sk, $\mathcal{S}_1$ cannot recover any useful information about the sensitive spatial data.

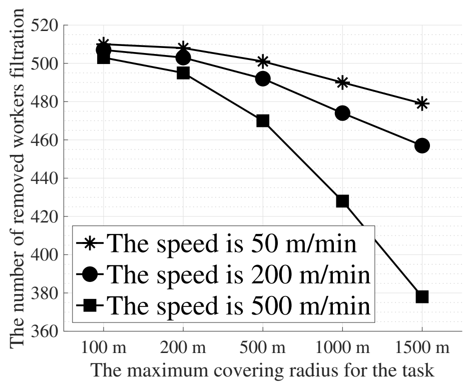
On the other hand, given sk, $\mathcal{S}_2$ can get the expression of Δ_5 based on Eq. (8.11). As we discussed in Section 8.2.2, the temporal information, the maximum covering radius, and workers' speeds can be considered as non-sensitive data as long as the spatial information are properly protected. Hence, the information $\Delta_1, \Delta_2, R_{\mathtt{t}}$ are known by $\mathcal{S}_2$. Consequently, for any of the two cases, $\mathcal{S}_2$ faces two unknown values (i.e., r_0 and $\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})$) in one equation and $\mathcal{S}_2$ cannot directly solve the



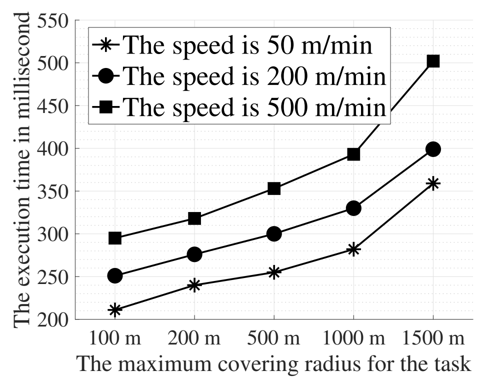
(a) The number of matched workers who meet the spatial-temporal constraints with $v_{w_i} = 200$ m/min, $R_{\mathtt{t}}$ varies from 100 m to 1500 m, and N_w varies from 500 to 2000.



(b) The number of matched workers who meet the spatial-temporal constraints with v_{w_i} varies 50 m/min to 500 m/min, $R_{\mathtt{t}}$ varies from 100 m to 1500 m, and N_w is 1000.



(c) The number of removed workers by the filtering process with v_{w_i} varies from 50 m/min to 500 m/min, $R_{\mathtt{t}}$ varies from 100 m to 1500 m, and N_w is 1000.



(d) The execution time (in millisecond) with v_{w_i} varies from 50 m/min to 500 m/min, $R_{\mathtt{t}}$ varies from 100 m to 1500 m, and N_w is 1000.

Figure 8.4: The experimental results of the proposed privacy-preserving worker selection scheme under different experimental settings.

equation. In other words, $\mathcal{S}_2$ cannot even recover $\mathtt{dist}^2(L_{\mathtt{t}}, L_{w_i})$, not to mention $L_{\mathtt{t}} = (x_{\mathtt{t}}, y_{\mathtt{t}})$ and $L_i = (x_i, y_i)$. In conclusion, in the proposed worker selection scheme, the participants' sensitive data (i.e., spatial information) are not disclosed to either of the two cloud servers.

8.5 Experimental Results

In this section, we evaluate the performance of the proposed worker selection scheme with a customized simulator built-in Java. Particularly, we simulate a sensing area with the size of 3000 m $\times$ 3000 m. We assume that $\mathcal{T}$ occurs at the center location $L_{\mathtt{t}} = (x_{\mathtt{t}}, y_{\mathtt{t}}) = (1500\text{ m}, 1500\text{ m})$. $\mathcal{T}$'s time duration $\tau_{\mathtt{t}}$ is set from 9:00 am to 11:00 am. The task's maximum covering radius $R_{\mathtt{t}}$ varies from 100 m, 200 m, 500 m, 1000 m to 1500 m. In addition, each worker w_i is randomly located in the sensing area, i.e., (x_i, y_i) are random integers in $[0, 3000]$. The time t_{w_i} is set to a random number between 8:00 am to 12:00 pm. We assume that the workers choose three different transportation modes, i.e., walking, taking a bike, and driving. Their corresponding speeds are 50 m/min, 200 m/min and 500 m/min, respectively. The number of total workers N_w is a changing variable with the values of 100, 200, 500, 1000, and 1500. In the experiment, we set the security parameters $k_0 = 2048, k_1 = 32$, and $k_2 = 160$. The experiment are conducted on a Windows 10 machine with an Intel(R) Core(TM) i7-6700 CPU @3.6-GHz, and 16GB RAM. All the experiments are run 10000 times and the average results are reported.

Fig. 8.4 (a) and (b) show the number of matched workers who meet both the spatial and temporal constraints under different experimental settings. In Fig. 8.4 (a), we set $v_{w_i} = 200$m/min, $R_{\mathtt{t}}$ varying 100 m to 1500 m, and N_w varying from 500 to 2000. In Fig. 8.4 (b), we set $N_w = 1000$, $R_{\mathtt{t}}$ varying 100m to 1500m, and v_{w_i} varying from 50m/min to 200m/min. From Fig. 8.4 (a) and (b), first we found that there is a significant positive correlation between the number of matched workers with $R_{\mathtt{t}}$, N_w, and v_{w_i}. The number of matched workers meeting the spatial-temporal constraint increases as the increase of the above three attributes. In addition, we can see that when $R_{\mathtt{t}} \leq 500\ m$, the number of matched workers is always less than 100 no matter what the values of N_{w_i} and v_{w_i} are. In some extreme cases (e.g., $R_{\mathtt{t}} = 100$ m, $v_{w_i} = 50$ m/min), no worker can be found to meet both the spatial and temporal constraints. Therefore, $\mathcal{R}$ needs to decide the reasonable $R_{\mathtt{t}}$ with the perspectives of the local population (e.g., N_w) and the average travelling speed in the local area (e.g., v_{w_i}). Fig. 8.4 (c) shows the number of removed workers by the proposed filtering process. We can see that the proposed filtering process can significantly improve the overall efficiency of the worker assignment in MCS services. In particular, nearly one-third to half of the total workers can be filtered out, which can significantly decrease the the number of encryption/decryption calculations and communications with two cloud servers. Fig. 8.4 (d) shows the execution time (in millisecond) under different experimental settings after the filtration process. The execution time is from 200 up to 500 milliseconds without considering the time cost during the communication. This result validates the efficiency of the proposed privacy-preserving worker selection scheme.

8.6 Related Work

Li et al. [57] proposed a multi-task allocation problem with temporal constraints. The proposed approach investigates the impact of temporal factors to multi-task allocation and is able to maximize the overall utility of the MCS platform. Moreover, two evolutionary algorithms are designed to solve the NP-complete problem in this work. Han et al. [38] studied the worker selection problem in MCS where the existing workers are fewer than the required number of participants. In their work, they proposed a hybrid framework to compensate for inadequate sensing opportunities by incentive mechanisms. In particular, in each sensing cycle, the proposed method combines mobile devices with static sensor nodes to generate uniformly distributed spatial-temporal data under the field coverage constraint. By utilizing calibration model for checking whether the workers' behaviors match the task's requirement, their method can balance sensing opportunities among different geographic regions and efficiently migrate workers from densely populated areas to sparsely populated areas. Zhang et al. [131] investigated the otptimal quality-aware coverage problem in MCS worker selection. By only selecting a subset of mobile workers, their proposed algorithm can maximize the coverage quality with constrained budget. Extensive simulations are conducted and the experimental results demonstrate the superiority of the proposed scheme over the baseline methods. Tao et al. [23] explored the worker selection problem with sensing-gain constraints in MCS. The authors first proposed a quality-aware worker reputation model with active factors. After that, they provided two types of sensing-gain constrained worker selection mechanism for both sufficient and insufficient user resources. Concretely, for the sufficient user resource scenario, they formulated the sensing-gain objective on recruiting cost and participant scale under constraints on data quality and task coverage. On the other hand, for the insufficient user resource scenario, they formulated the sensing-gain objective on data quality. Comprehensive simulations over real-world data have been conducted. The evaluation results showed that the proposed worker reputation model can distinguish high-quality workers with different active levels. Besides, experimental analyses verify that their algorithm can effectively select qualified workers with ideal recruitment budget and guaranteed data quality.

8.7 Concluding Remarks

In this chapter, we have described a practical MCS worker selection scenario where the selected workers should meet both the spatial and temporal constraints. To protect the participants' (i.e., both the task requestor and the worker) sensitive personal data, we proposed a privacy-preserving worker selection scheme based on the symmetric homomorphic encryption (SHE) approach, which can protect participants' spatial information from being disclosed while still enabling effective worker assignment. In addition, by applying a pre-filtering process, the number of considered candidates can be significantly reduced. With extensive experimental evaluations, we demonstrate the effectiveness of our solution. For future work, we will work on privacy-preserving worker selection over workers' trajectories.

Chapter 9

Worker Selection in MCS based on Spatial-temporal Constraint Over Tentative Future Trajectories

9.1 Introduction

In the previous two chapters, we proposed privacy-preserving worker selection schemes form the perspectives of both spatial-temporal similarity and spatial-temporal constraints. By extending the problems in the previous two chapters, in this chapter, we investigate a more realistic scenario for MCS worker selection based on the spatial-temporal constraint over future trajectories. Essentially, we postulate that each worker will usually plan a future trajectory (i.e., a sequence of daily activities) in advance. The worker may want to travel to the task location and perform the task between two consecutive activities only if the task location is reachable and the extra travel cost does not affect his/her original schedule. In other words, the worker is only willing to accept the tasks that meet his/her spatial-temporal constraint. Next, a real-world example is presented to describe the motivational scenario in Fig. 9.1.

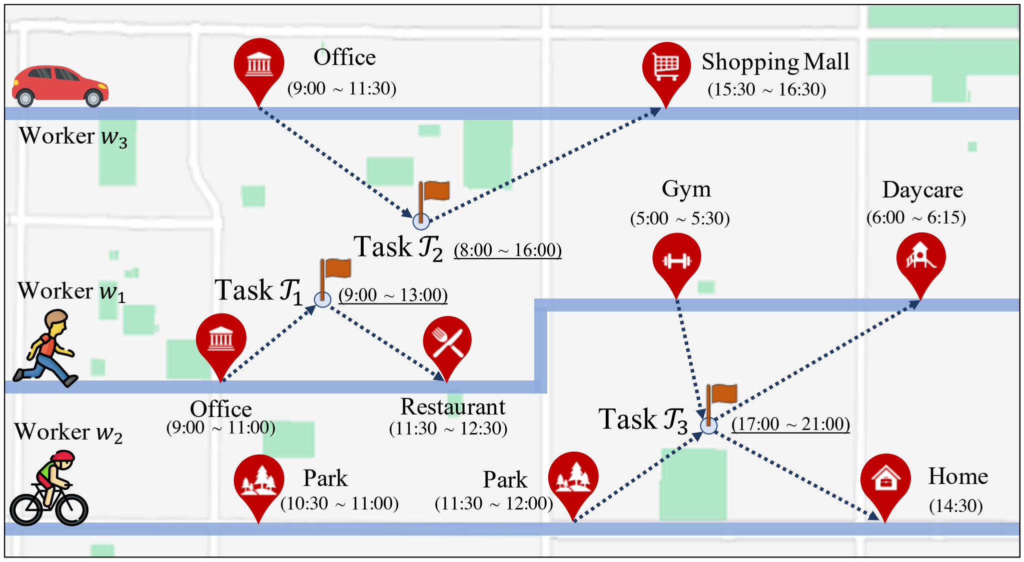


Figure 9.1: An example of spatial-temporal constraint based worker selection over a sequence of activities.

In Fig. 9.1, there are three MCS tasks $\{\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3\}$ and three workers $\{w_1, w_2, w_3\}$. Each task is associated with a specific location and a valid time duration, e.g., $\mathcal{T}_1$ is to monitor the air quality of a local industry zone from 9:00 to 13:00. Each worker plans a sequence of daily activities in advance. For instance, w_1 will arrive at and leave the office at 9:00 and 11:00. Then, w_1 has an reservation at a nearby restaurant to have launch with his/her colleague at 11:30 and the lunch is supposed to be finished within an hour. Therefore, w_1 can be considered as a candidate for $\mathcal{T}_1$ since after leaving the office at 11:00, w_1 has 30 mins free time to travel to $\mathcal{T}_1$'s location to perform the task[1] (during $\mathcal{T}_1$'s valid time duration 9:00 to 13:00) and then go to the restaurant before 11:30. Similarly, w_1 and w_3 can be selected for $\mathcal{T}_3$ and $\mathcal{T}_2$, respectively. Notably, w_2 will not be assigned for $\mathcal{T}_3$ since he/she cannot arrive at $\mathcal{T}_3$ during the task's valid time duration (17:00 to 21:00). Besides, workers' daily activities are planned in advance. It is possible that the workers do not exactly follow the reported activities due to some unexpected or emergency reasons. Hence, even though a worker has been selected, he/she may not be able to perform the task.

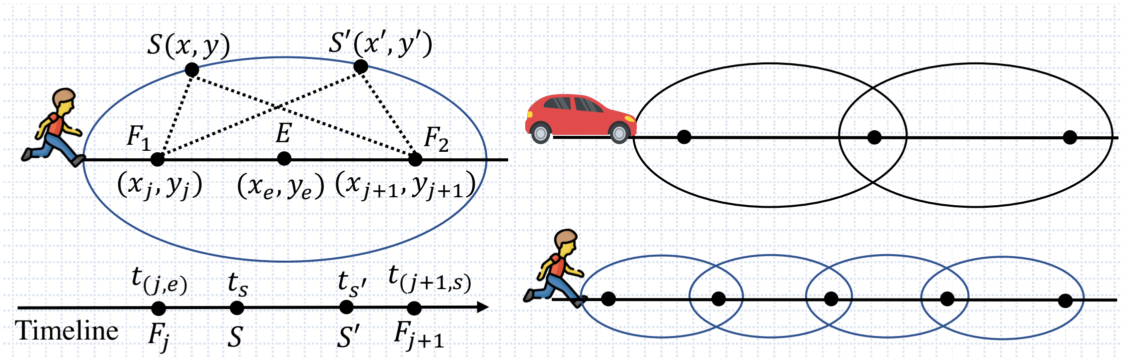
[1]For simplicity sake, the tasks considered in this work are easy (e.g., taking a picture, checking available parking spots) and can be finished without any sojourn time.

Finally, the MCS platform wants to select suitable workers for each MCS task, while considering both the reliability of the selected worker and the overall budget at the same time.

It is worth noting that a worker's future trajectory contains sensitive personal data, e.g., real-time spatial-temporal information, which should be protected in the process of worker selection. Consequently, the workers may not want to reveal their actual future locations and the corresponding arriving/leaving times. In this work, aiming at solving the problem in above example in Fig. 1, we propose a novel and realistic MCS worker selection scenario based on workers' tentative future trajectories. Our contributions can be summarized as follows.

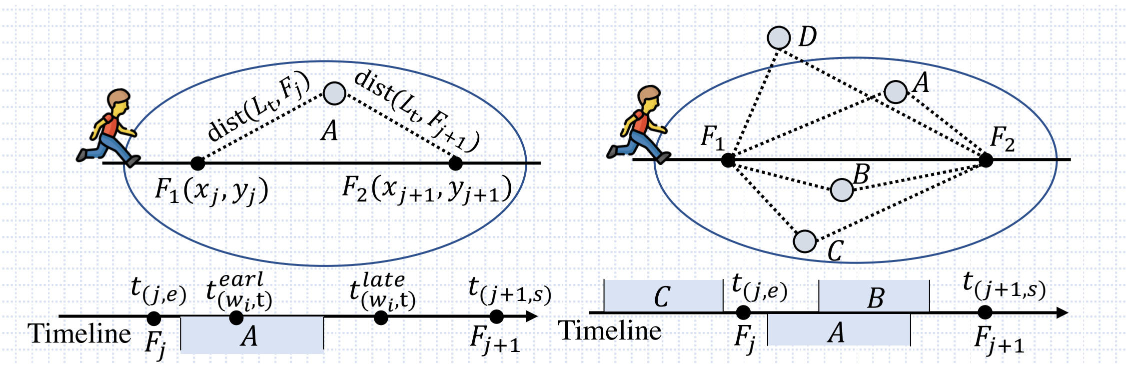
- First, we introduce a novel worker selection problem and formulate workers' spatial-temporal constraints based on their future trajectories.
- Second, we devise a hybrid approach for indexing workers' future trajectories and designing a greedy algorithm to maximize reliability score.
- Last, we propose a novel privacy-preservation approach based on restricted spatial-temporal constraints. In particular, instead of submitting accurate information, each worker sends the MCS platform his/her restricted spatial-temporal information. With the restricted information, the MCS platform can still select workers based on spatial-temporal constraints..

The roadmap of this chapter is organized as follows. In Section 9.2, we formally define the problem in this chapter. In Section 9.3, we describe the proposed scheme including the spatial-temporal constraint matching and the greedy worker selection approach. In Section 9.4, we evaluate the experimental results for the proposed scheme. Then, we discuss some related work in Section 9.5. Finally, we conclude the summary of this chapter in Section 9.6.



(a) The worker's spatial coverage area between $o_j = \langle F_j, \tau_j \rangle$ and $o_{j+1} = \langle F_{j+1}, \tau_{j+1} \rangle$ forms an ellipse. t_S and $t_{S'}$ are the arrival times for locations S and S'.

(b) Two workers and their future trajectories in forms of ellipses. The speed of the car is much faster than the walking man, so the car's ellipse is larger than the man's ellipse.



(c) The worker's earliest arrival time $t^{earl}_{(w_i,\text{t})}$ and latest departure time $t^{late}_{(w_i,\text{t})}$. The worker meets the temporal constraint with task A.

(d) Only ask A and B meet both the spatial and temporal constraints. Task D does not meet the spatial constraint, and task C does not meet the temporal constraint.

Figure 9.2: The illustrations of workers' spatial coverage, future trajectories, and the spatial-temporal constraint.

9.2 Problem Formulation

In this section, we formulate the problem of spatial-temporal constraint based worker selection over future trajectories.

Definition 9.2.1. ***(The MCS Task)*** *An MCS task $\mathcal{T}$ includes a location L_t and a temporal duration τ_t. For simplicity sake, we consider $L_t = (x_t, y_t)$ is a point* *in the 2D space;* $\tau_t = (t_{(t,s)}, t_{(t,e)})$ *where* $t_{(t,s)}$ *and* $t_{(t,e)}$ *are the start time and end time for the task, respectively. Besides, each task* $\mathcal{T}$ *is associated with a budget* B_t*, which specifies the maximum monetary incentives provided for the selected workers. Basically,* $\mathcal{T}$ *can be denoted as* $\mathcal{T} = \langle L_t, \tau_t, B_t \rangle$.

Definition 9.2.2. ***(Worker)*** *A worker* w_i *is a participant who carries smart devices and is willing to accept certain MCS tasks between two consecutive locations that s/he expects to be at, in future. To be selected,* w_i *needs to submit his/her future trajectory* $\mathbb{T}_i$ *(will be defined next) to the platform. Besides,* w_i *is associated with a trustability* T_i*, which indicates the reliability of the provided information. In addition, since different workers may choose different transportation modes (e.g., walk, take a bike, drive), each worker* w_i *has a predefined maximum traveling speed* v_i*. Essentially,* w_i *can be denoted as* $w_i = \langle \mathbb{T}_i, T_i, v_i \rangle$.

Definition 9.2.3. ***(Worker's Future Trajectory)*** *A worker* w_i*'s future trajectory* $\mathbb{T}_i$ *is defined as a sequence of finite, time-ordered daily activities, i.e.,* $\mathbb{T}_i = \{o_1, o_2, \ldots, o_n\}$[2]. *Each activity* o_j *for* $j \in [1, n]$ *includes the location information* $F_j = (x_j, y_j)$ *and the temporal information* $\tau_j = (t_{(j,s)}, t_{(j,e)})$*, such that* w_i *should arrive at* F_j *at time* $t_{(j,s)}$ *and leave* F_j *at time* $t_{(j,e)}$*, i.e.,* $o_j = \langle F_j, \tau_j \rangle$.

Next, for worker w_i, we need to find his/her furthest travel distance between two reported activities. Then, we can formulate w_i's maximum spatial coverage area. In Fig. 9.2 (a), w_i's two consecutive activities are $o_j = \langle F_j, \tau_j \rangle$ and $o_{j+1} = \langle F_{j+1}, \tau_{j+1} \rangle$. We assume that S and S' are any two furthest reachable locations that w_i can arrive between o_j and o_{j+1}. Let $\Delta t_j = t_{(j+1,s)} - t_{(j,e)}$. Basically, w_i can follow the path $F_jS \rightarrow SF_{j+1}$ to reach F_{j+1} from F_j; w_i can also reach F_{j+1} by the path $F_jS' \rightarrow S'F_{j+1}$, as long as the travel time is less than Δt_j. Given w_i's maximum

[2]In fact, $\mathbb{T}_i$ does not necessarily to be a continuous sequence of observations. It can some individual ellipses that have no overlap with each other).

travel speed v_i, we have

$$\Delta t_j \times v_i = \texttt{dist}(F_j, S) + \texttt{dist}(S, F_{j+1}) = \texttt{dist}(F_j, S') + \texttt{dist}(S', F_{j+1}), \qquad (9.1)$$

where $\texttt{dist}(p, q)$ is the Euclidean distance between two points p and q in the 2D space. Essentially, any location that w_i can reach between o_j and o_{j+1} satisfies Eq. (9.1). Thus, all the possibly furthest locations between o_j and o_{j+1} form an ellipse [105] (see in Fig. 9.2 (a)). More details about the ellipse between two consecutive activities can be found in Appendix A.

Definition 9.2.4. ***(Worker's Spatial Coverage Area)*** *A worker w_i's spatial coverage area $\boldsymbol{\mathit{Area}}_{(w_i,o_j,o_{j+1})}$ includes all w_i's possible locations in-between any two consecutive activities o_j and o_{j+1} from $\mathbb{T}_i$. The shape of $\boldsymbol{\mathit{Area}}_{(w_i,o_j,o_{j+1})}$ is an ellipse with the equation:*

$$(x - x_e)^2/a^2 + (y - y_e)^2/b^2 = 1. \qquad (9.2)$$

Here, (x_e, y_e) is the center of the ellipse and (a, b) are the semi-major and semi-minor axes of the ellipse [138], such that

$$\begin{aligned} x_e &= \frac{x_j + x_{j+1}}{2}, \quad y_e = \frac{y_j + y_{j+1}}{2}, \quad a = \frac{v_i \times (t_{(j+1,s)} - t_{(j,e)})}{2} \\ b &= \frac{\sqrt{v_i^2 \times (t_{(j+1,s)} - t_{(j,e)})^2 - (x_{j+1} - x_j)^2 - (y_{j+1} - y_j)^2}}{2}. \end{aligned} \qquad (9.3)$$

The size of the ellipse varies based on each worker's maximum traveling speed, as shown in Fig. 9.2 (b).

Definition 9.2.5. ***(The Spatial Constraint)****: Given an MCS task $\mathcal{T} = \langle L_t, \tau_t, B_t\rangle$ and a worker w_i's future trajectory $\mathbb{T}_i$, we say w_i meets the spatial constraint with $\mathcal{T}$ (denoted by $w_i \cap_s \mathcal{T} \neq \emptyset$) if there exists a pair of consecutive activities $o_j = \langle F_j, \tau_j\rangle$*

and $o_{j+1} = \langle F_{j+1}, \tau_{j+1} \rangle$ *from* $\mathbb{T}_i$*, such that*

$$\boldsymbol{dist}(L_t, F_j) + \boldsymbol{dist}(L_t, F_{j+1}) \leq 2 \times a. \tag{9.4}$$

Definition 9.2.6. *(**The Temporal Constraint**): Given the MCS task* $\mathcal{T} = \langle L_t, \tau_t, B_t \rangle$ *and a worker* w_i*'s future trajectory* $\mathbb{T}_i$*, for any pair of consecutive activities* $o_j = \langle F_j, \tau_j \rangle$ *and* $o_{j+1} = \langle F_{j+1}, \tau_{j+1} \rangle$ *from* $\mathbb{T}_i$*, we can calculate* w_i*'s earliest arrival time* $t^{earl}_{(w_i,t)}$ *and latest departure time* $t^{late}_{(w_i,t)}$ *(see in Fig. 9.2 (c)) at location* L_t *as:*

$$t^{earl}_{(w_i,t)} = t_{(j,e)} + \frac{\boldsymbol{dist}(L_t, F_j)}{v_i}, \; t^{late}_{(w_i,t)} = t_{(j+1,s)} - \frac{\boldsymbol{dist}(L_t, F_{j+1})}{v_i}. \tag{9.5}$$

Essentially, if the time duration $(t^{earl}_{(w_i,t)}, t^{late}_{(w_i,t)})$ *overlaps the task's time duration* $\tau_t = (t_{(t,s)}, t_{(t,e)})$*, we say* w_i *meets the temporal constraint with the task* $\mathcal{T}$*, denoted by* $w_i \cap_t \mathcal{T} \neq \emptyset$*.*

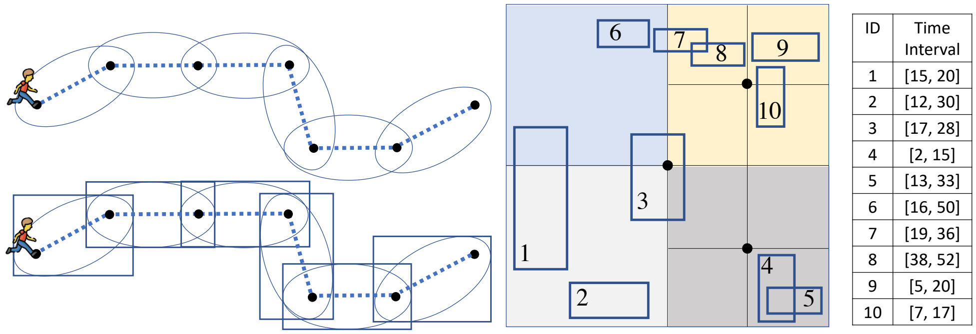
Definition 9.2.7. *(**Worker's Task Cost**) The task requestor or the MCS platform needs to provide some incentives to each selected worker, as each worker's remuneration. In this chapter, we assume that for worker* w_i*, his/her task cost* c_i *equals to the overall travel distance taken for performing the task [33]. Hence, we have*

$$c_i = \boldsymbol{dist}(L_t, F_j) + \boldsymbol{dist}(L_t, F_{j+1}). \tag{9.6}$$

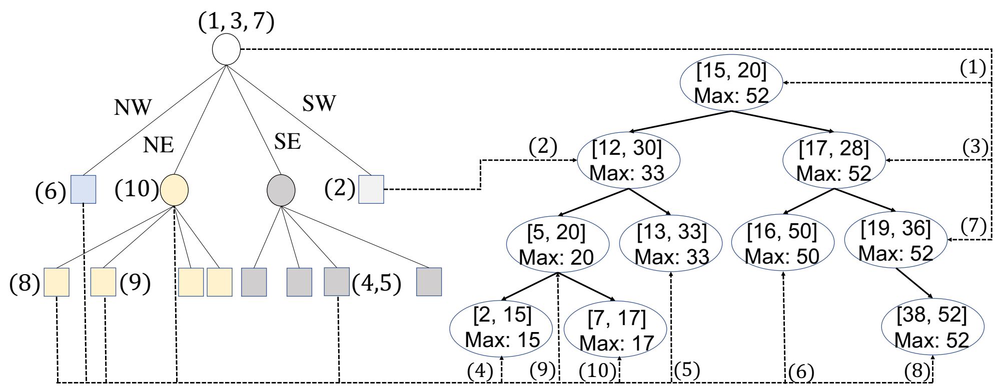
Definition 9.2.8. *(**The Reliability of Worker Selection**): Given an MCS task* $\mathcal{T}$ *and a group of selected workers* $\mathcal{W}_s = \{w_1, w_2, \ldots, w_k\}$*. The reliability of worker selection* $\boldsymbol{Rea}_{(\mathcal{T},\mathcal{W}_s)}$ *is defined as:*

$$\mathsf{Rea}_{(\mathcal{T},\mathcal{W}_s)} = 1 - \prod_{\forall w_i \in \mathcal{W}_s} (1 - T_i), \tag{9.7}$$

where T_i *is* w_i*'s trustability for* $i \in [1, k]$*, and can be measured based on historical*



(a) A worker's future trajectory consists of a sequence of ellipses. Each ellipse is approximated by a rectangle. Each rectangle is associated with the spatial information (see the decomposed regions) and the temporal information (see the table of time intervals).



(b) The MX-CIF quadtree (left) and the interval tree (right) are built for storing each rectangle's spatial and temporal information, respectively. NW: north west, NE: north east, SE: south east, SW: south west. The dotted line connect the nodes in interval tree with the nodes in quadtree.

Figure 9.3: The illustrations our proposed scheme for indexing the spatial and temporal information.

sensing records [137]. Specifically, $\prod_{\forall w_i \in \mathcal{W}_s}(1 - T_i)$ means the probability that all the selected workers cannot perform the task $\mathcal{T}$. Hence, $1 - \prod_{\forall w_i \in \mathcal{W}_s}(1 - T_i)$ is the probability that there exists at least one worker who can conduct $\mathcal{T}$.

Except spatial-temporal constraint, the overall task costs cannot exceed the task budget, and the task' reliability should be maximized. The MCS worker selection problem over future trajectories is formulated as follows.

Given an MCS task $\mathcal{T} = \langle L_{\mathtt{t}}, \tau_{\mathtt{t}}, B_{\mathtt{t}} \rangle$ and a large group of workers $\mathcal{W}$, the problem of spatial-temporal constraint based worker selection is to assign $\mathcal{T}$ with a set of workers $\mathcal{W}_s \subseteq \mathcal{W}$, we have:

- for $\forall w_i \in \mathcal{W}_s$, w_i meets both the spatial constraint and temporal constraint, i.e., $w_i \cap_s \mathcal{T} \neq \emptyset$ and $w_i \cap_t \mathcal{T} \neq \emptyset$;
- the reliability of worker selection $\mathtt{Rea}_{(\mathcal{T},\mathcal{W}_s)}$ is maximized;
- the overall task cost should be smaller than the task budget, i.e., $\sum_{w_i \in \mathcal{W}_s} c_i \leq B_{\mathtt{t}}$.

9.3 The Proposed Scheme

In this section, we describe our proposed scheme including privacy-preserving worker selection, spatial-temporal constraint matching, and greedy worker selection.

9.3.1 Privacy-preserving Worker Selection

As we mentioned before, workers' future trajectories may contain sensitive personal information, which should be protected during the process of worker selection. Therefore, we design a privacy-preserving worker selection approach based on restricted spatial-temporal constraints. In this approach, instead of submitting accurate real-time spatial-temporal information, each worker provides restricted spatial-temporal information to the MCS platform, depending on which, the MCS platform can effectively select qualified workers without knowing any sensitive information. The details of the privacy-preservation technique are discussed next.

9.3.1.1 Restricted Spatial Coverage Area

A worker w_i defines the restricted spatial coverage area $\mathtt{Area}^r_{(w_i,o_j,o_{j+1})}$ as the inscribed circle of the ellipse in his/her future trajectory $\mathbb{T}_i$ (as shown in Fig. 9.4 (a)).

Upon receiving $\texttt{Area}^r_{(w_i,o_j,o_{j+1})}$, the MCS platform cannot retrieve w_i's accurate spatial information since given a circle, there may be infinite number of circumscribed ellipses. It is worth noting that to preserve the privacy of the sensitive spatial information, the worker w_i needs to sacrifice some space coverage. As seen in Fig. 9.4 (b), the shadow part is the area that is covered by the real spatial coverage area $Area_{(w_i,o_j,o_{j+1})}$, but not covered by the restricted spatial coverage area $\texttt{Area}^r_{(w_i,o_j,o_{j+1})}$. The sacrifice degree $\texttt{Ratio}^r_s$ can be represented by the ratio between $\texttt{Area}^r_{(w_i,o_j,o_{j+1})}$ and $Area_{(w_i,o_j,o_{j+1})}$, i.e.,:

$$\begin{aligned}\texttt{Ratio}^r_s &= \frac{\texttt{Area}^r_{(w_i,o_j,o_{j+1})}}{\texttt{Area}_{(w_i,o_j,o_{j+1})}} = \frac{\pi \times b^2}{\pi \times a \times b} \\ &= \frac{\sqrt{v_i^2 \times (t_{(j+1,s)} - t_{(j,e)})^2 - (x_{j+1} - x_j)^2 - (y_{j+1} - y_j)^2}}{v_i \times (t_{(j+1,s)} - t_{(j,e)})}.\end{aligned} \tag{9.8}$$

However, in this work, we consider a dense MCS scenario in which there are a large number of workers waiting for the task assignment. Therefore, even though each single worker loses some spatial coverage, the overall task allocation does not be affected. Besides, since the restricted spatial coverage area $\texttt{Area}^r_{(w_i,o_j,o_{j+1})}$ meets the spatial constraint, w_i should also meet the spatial constraint.

9.3.1.2 Restricted Temporal Constraint

Similarly, instead of submitting accurate temporal information, w_i sends a restricted time duration $(t^{\texttt{earl}}_{(w_i,F_c)}, t^{\texttt{late}}_{(w_i,F_c)})$ to the MCS platform to indicate his/her earliest arrival time and latest departure time at the center of the restricted spatial coverage area F_c. After receiving the restricted time duration, the MCS platform can calculate the restricted earliest arrival time $t^{\texttt{earl-r}}_{(w_i,\texttt{t})}$ and restricted latest departure time $t^{\texttt{late-r}}_{(w_i,\texttt{t})}$ $\texttt{t}$ as

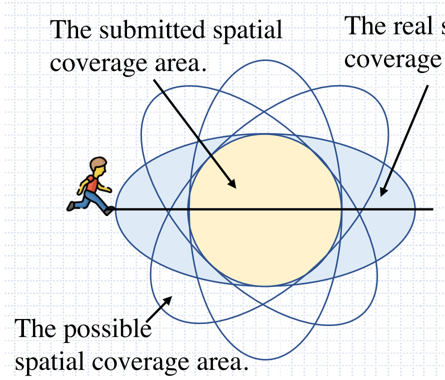
$$\begin{aligned}t^{\texttt{earl-r}}_{(w_i,\texttt{t})} &= t_{(j,e)} + \frac{\texttt{dist}(L_\texttt{t}, F_j) + \texttt{dist}(F_j, F_c)}{v_i}, \\ t^{\texttt{late-r}}_{(w_i,\texttt{t})} &= t_{(j+1,s)} - \frac{\texttt{dist}(L_\texttt{t}, F_{j+1}) + \texttt{dist}(F_{j+1}, F_c)}{v_i}.\end{aligned} \tag{9.9}$$

Based on Eq. (9.5) and (9.9), it is easy to know that $t^{\texttt{earl-r}}_{(w_i,\texttt{t})} > t^{\texttt{earl}}_{(w_i,\texttt{t})}$ and $t^{\texttt{late-r}}_{(w_i,\texttt{t})} < t^{\texttt{late}}_{(w_i,\texttt{t})}$. Therefore, if a worker w_i meets the temporal constraint based on $(t^{\texttt{earl-r}}_{(w_i,\texttt{t})}, t^{\texttt{late-r}}_{(w_i,\texttt{t})})$, he/she should meet the temporal constraint based on $(t^{\texttt{earl}}_{(w_i,\texttt{t})}, t^{\texttt{late}}_{(w_i,\texttt{t})})$.

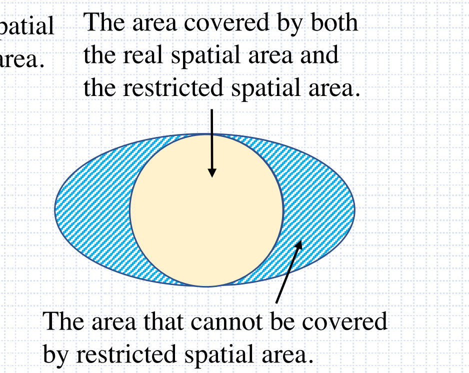
Finally, w_i prepares a sequence of restricted spatial coverage area $\texttt{Area}^r_{(w_i,o_j,o_{j+1})}$ and their corresponding restricted time duration $(t^{\texttt{earl}}_{(w_i,F_c)}, t^{\texttt{late}}_{(w_i,F_c)})$ and send them to the MCS platform for worker selection (as seen in Fig. (9.4)). Note that the MCS platform can provide the workers to automate the preparation of these sequence of circles by providing an interactive visual tool. Essentially, the restricted spatial-temporal information does not reveal the accurate sensitive information of each worker, while still enabling the process of worker selection based on spatial-temporal constraints. For instance, there will be many different possible real trajectories given one sequence of circles (as see in Fig. (9.4)).

9.3.2 Spatial-temporal Constraint Matching

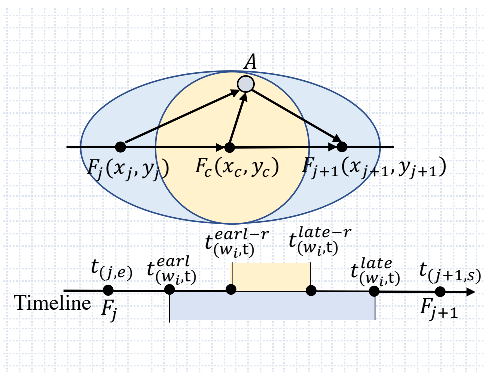
After receiving all workers' submitted information, the MCS platform should first find the workers who meet both the spatial and temporal constraints. Due to workers' highly dynamic mobility, recruitment decisions should be made as fast as possible [41]. In this paper, we design a novel hybrid index for facilitating the speed of worker selection. Essentially, the proposed hybrid index can represent both workers' original trajectory information (a sequence of ellipses) and their restricted spatial-temporal information (a sequence of circles). Taking the original trajectory as an example, specifically, for any worker w_i, the platform will first split $\mathbb{T}_i$ into a sequence of ellipses (see in Definition 9.2.4). Next, a rectangle is used to approximate each ellipse (see Fig. 9.3 (a)). After that, a MX-CIF quadtree I_Q [92] and a interval tree I_T [81] (see Fig. 9.3 (b)) are built to store each rectangle's spatial and temporal information, respectively. Given a task query $\mathcal{T} = \{L_{\texttt{t}}, \tau_{\texttt{t}}, B_{\texttt{t}}\}$ where $L_{\texttt{t}} = (x_{\texttt{t}}, y_{\texttt{t}})$ and $\tau_{\texttt{t}} = (t_{(\texttt{t},s)}, t_{(\texttt{t},e)})$, the platform queries the rectangles from I_Q based on $L_{\texttt{t}}$ and



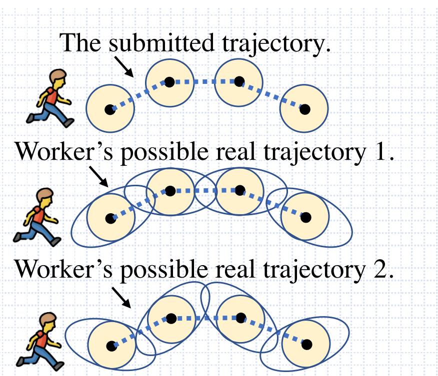
(a) The restricted spatial coverage (the light gold circle) submitted by the worker, which does not disclose any information about the accurate spatial locations (the light blue ellipse).



(b) Illustration of the relationship between the real spatial coverage area and the restricted spatial coverage area, where the restricted spatial coverage area is part of real spatial coverage area.



(c) The worker's restricted earliest arrival time $t^{earl-r}_{(w_i,t)}$ and the restricted latest departure time $t^{late-r}_{(w_i,t)}$. The restricted time duration $(t^{earl-r}_{(w_i,t)}, t^{late-r}_{(w_i,t)})$ is part of the real time duration.



(d) Illustration of the worker's real trajectory and the corresponding submitted trajectory based on the restricted spatial-temporal information. There are many different possible trajectories given one submitted trajectory.

Figure 9.4: Illustration of workers' restricted spatial and temporal information.

the rectangles from I_T based on $\tau_{\mathtt{t}}$. Finally, the intersection of rectangles from both I_Q and I_T are the final candidates for checking the spatial and temporal constraints. The approximate rectangles can also used for representing circles in the restricted trajectories.

9.3.3 The Greedy Worker Selection Algorithm

After finding all the workers who meet the spatial-temporal constraint, we need to select a group of qualified workers $\mathcal{W}_s$ to maximize the reliability of worker selection while keeping the overall task cost smaller than the task budget. Hence, we need to consider the trade-off between workers' trustability and their task costs. Here, we provide the definition of *Reliability Score*, which is used as the essential idea for building a greedy worker selection algorithm.

Definition 9.3.1. *(**Reliability Score**) Let $\mathcal{W}_s$ denote a group of workers that are currently being selected. If w_i is the next worker to be selected, then his/her reliability score $\widetilde{c}_i$ is the marginal reliability gain per unit cost. Basically, we define*

$$\widetilde{c}_i = \Delta \boldsymbol{Rea}/c_i = (\boldsymbol{Rea}_{(\mathcal{T},\mathcal{W}_s \cap w_i)} - \boldsymbol{Rea}_{(\mathcal{T},\mathcal{W}_s)})/c_i. \tag{9.10}$$

Each time when we want to select a new worker w_i for an MCS task, we aim to recruit the worker who has the highest reliability score value $\widetilde{c}_i$. Equivalently, this new worker can maximize the reliability of worker selection with the lowest task cost. And this is the fundamental idea of the proposed greedy worker selection algorithm. This process will be repeated until the sum of all the workers' task costs reach the limits of the task budget. The details of the greedy approach are shown in Algorithm 11.

9.4 Experimental Results

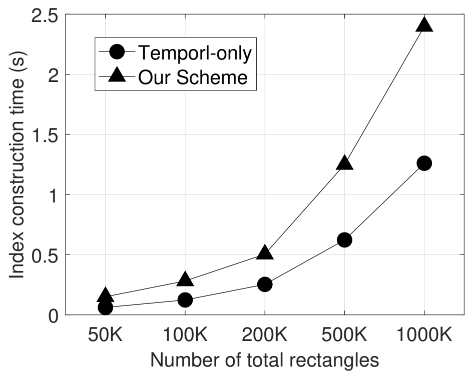
In this work, we simulate an MCS environment as follows. (1) We assume that the workers choose three different transportation modes, i.e., walking, taking a bike, and driving. Their corresponding average speeds are $50m/min, 200m/min$, and $500m/min$. Besides, the total number of workers $N_{\texttt{worker}}$ various from $5K$, $10K$, $20K$,

Algorithm 11 The Greedy Worker Selection Algorithm

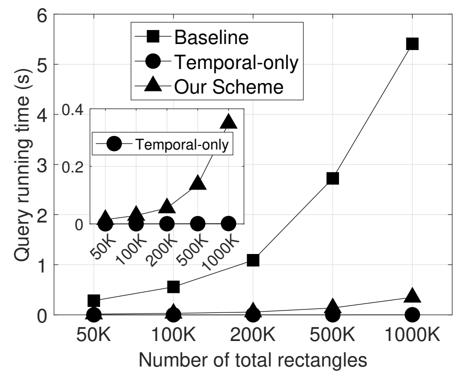
Input: The group of all the workers $\mathcal{W}$ who meets the spatial-temporal constraint with an MCS task $\mathcal{T}$
Output: The subset of selected workers $\mathcal{W}_s$
Set $\mathcal{W}_s = \emptyset$, $all\_cost = 0$;
Store all the workers in $\mathcal{W}$ into a Max Priority Queue $\mathcal{Q}$ based on workers' reliability score $\widetilde{c}_i$;
while $\mathcal{Q}$ is not empty **do**
  Let w_f denote the first worker in $\mathcal{Q}$, and c_f is w_f's travel cost;
  if $c_f + \text{all\_cost} < B_{\mathtt{t}}$ **then**
    $\mathcal{W}_s \leftarrow \mathcal{W}_s \bigcup w_f$, $all\_cost \leftarrow all\_cost + c_f$
    Remove w_f from $\mathcal{Q}$;
  end if
end while
return The group of selected workers $\mathcal{W}_s$

$50K$, and $100K$. We assume that for any worker w_i, there are 11 daily activities in the future trajectory, i.e., $\mathbb{T}_i = \{o_1, o_2, \ldots, o_{11}\}$. Consequently, there are 10 ellipses in each trajectory, and the total number of ellipses (e.g., rectangles) $N_{\mathtt{rect}}$ are varied as $50K, 100K, 200K, 500K$, and $1000K$. The percentages of workers among $N_{\mathtt{worker}}$ using three transportation models are 45%, 45%, 10%. (2) We stimulate a sensing area with the overall size of $30km \times 30km$, where the bottom-left coordinates are (0, 0) and the top-right coordinates are (30K, 30K). For each worker w_i, we assume that the time interval between any two continuous daily activities o_j and o_{j+1} is a random number in the range of $(15min, 45min)$. The location of the first daily activity $F_1 = (x_1, y_1)$ is a random point in the overall sensing area, i.e., $x_1, y_1 \in (0, 30K)$. The worker can move in any direction, and the location of the next daily activity can be calculated by Eq. (9.3). All the locations of workers' daily activities should be located in the sensing area. (3) To evaluate the performance of our proposed index, we implement two methods for comparison purpose. One is a **baseline method**, where there is no data structure for indexing the spatial and temporal information. Specifically, the baseline method checks each rectangle from the database one by one and determines if it meets both the spatial and temporal constraints. Another one is a **temporal-only method** based on [96]. Specifically, each day is split into 12 times slots and each rectangle is stored in the minimal temporal grid that can fully enclose it. In addition, the task' location $L_{\mathtt{t}}$ is a random point in the sensing area, and the task's time duration $\tau_{\mathtt{t}}$ various from $30min$ to $120min$. All the experiments are implemented with Python programming language on a machine with an Intel(R) Core(TM) i7-6700 CPU @3.6-GHz and Windows 10 operating system. We conduct each experiment for 1000 times and the average results are reported. It is worth noting that our current experiments are conducted based on workers' real trajectory information (i.e., a sequence of ellipses). The same evaluation process can also be applied to the restricted spatial-temporal information (i.e., a sequence of circles) since the approximate rectangles can also represent the spatial information in terms of circles.

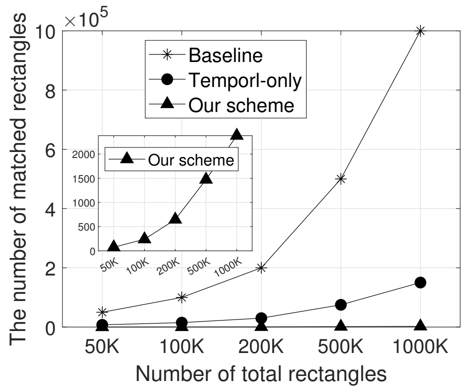
Fig. 9.5 shows the experimental results. Fig. 9.5 (a) presents the index construction time under different $N_{\mathtt{rect}}$. It is very quick to build the indexes. When $N_{\mathtt{rect}} = 1000K$, it takes only about $1.5s$ and $2.5s$ for the temporal-only method and our scheme, respectively. From Fig. 9.5 (b), we can see that comparing with the baseline method, the querying time for the temporal-only method and our approach are both quite fast. In Fig. 9.5 (c), we can see that our scheme is much more effiective in term of finding the matching results under different $N_{\mathtt{rect}}$. It can find around 2000 candidates when $N_{\mathtt{rect}} = 1000K$, whereas the number for the temporal-only method is around 2×10^5. Fig. 9.5 (d) evaluates the number of matched candidates for the proposed scheme versus different query time intervals. It is clear to see that the number of matched rectangles increases as the time interval and $N_{\mathtt{rect}}$ increase.



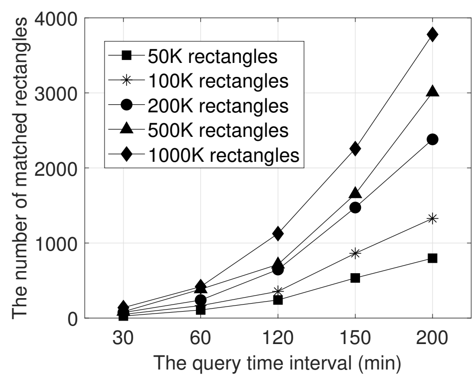
(a) The running time (s) for building the MX-CIF quadtree and the interval tree with different $N_{\mathtt{rect}}$.



(b) The query time (s) based on baseline, the MX-CIF quadtree, and the interval tree with different $N_{\mathtt{rect}}$.



(c) The number of matched rectangles after different querying methods under diffrent $N_{\mathtt{rect}}$.



(d) The number of matched rectangles for the proposed scheme given different query time intervals.

Figure 9.5: The experimental results in terms of the time for building the trees, the time for querying the matched rectangles, and the number of matched candidates under different approaches.

9.5 Related Work

In this section, we briefly introduce some major research problems in MCS based on trajectory and mobility analyses. Chen et al. [15] studied the problem of how to choose the most appropriate workers when the budget used to pay the workers is limited. They proposed a novel coverage assessment in MCS worker selection considering both uniform coverage and maximum coverage. Then, they designed a trajectory segment selection scheme. Instead of selecting the whole trajectory of a worker, the proposed scheme can choose certain trajectory segments. Their experimental evaluation is conducted in both offline and online manner on a read dataset. The results prove the effectiveness and the advantage of the proposed scheme regarding the coverage quality. Wang et al. [110] proposed an efficient vehicle recruitment framework in MCS based on the prediction of vehicles' mobility trajectories. In particular, they evaluate two types of mobility models for vehicles, namely, deterministic and probabilistic models. For the deterministic trajectory model, the authors designed an LP-relaxation-based heuristic algorithm to minimize the overall recruitment cost. For the probabilistic trajectory model, they proposed a greedy algorithm with a guaranteed approximation ratio. Niu et al. [78] designed a novel system model and optimize the selection of public transports based on their predictable trajectories. In their work, the public transport is defined as the working unit embedded with substantial sensors. But different from the intelligent terminals or human participant, the trajectory of public transports is predictable, and can be used to achieve high-quality sensing services. They first prove that public transport selection is non-deterministic polynomial-time hardness. Then, they propose an approximation algorithm to make the coverage close to 1. Finally, they evaluate the proposed approach with samples of real T-Drive trajectory dataset. Based on the experimental results, their algorithm can achieve a near optimal coverage and outperform the baseline methods.

9.6 Concluding Remarks

In this chapter, we described a novel MCS worker selection problem based on workers' tentative future trajectories. We presented a hybrid indexing approach to represent workers' spatial-temporal information and a greedy algorithm that attempts to maximize workers' reliability score. Furthermore, we propose a privacy-preserving worker selection approach to protect workers' sensitive information from being disclosed. In particular, instead of submitting accurate information, workers send their approximate spatial-temporal coverage for spatial-temporal matching. With extensive experimental results we demonstrate the effectiveness of our solution. There are many interesting future directions based on the preliminary study of this work, e.g., worker selection based on future trajectories on road networks, privacy preservation of workers' future trajectories, worker selection in terms of task trajectories, and the evaluation of more sophisticated indexing structures using real-world data.

Chapter 10

Conclusion and Future Work

In this chapter, we summarize the major contributions of this thesis and propose some potential future research directions.

10.1 Contributions

The major contributions of this thesis can be summarized as follows.

- First, we have proposed a privacy-preserving worker selection scheme for MCS applications based on (probabilistic) skyline computation. Our proposed scheme can efficiently select suitable and reliable workers without revealing workers' personal information. Specifically, the privacy-preserving probabilistic skyline is used to calculate workers' *PS-score* based on historical reviews. *PS-score* reveals the fluctuation of workers' past performance and can be used as their trustability. Then, skyline computation is used for selecting workers in terms of ask price, task similarity, and trustability. A non-interactive encrypted integer comparison protocol called *NIEC* is designed for supporting our scheme, which can securely compare two integers in a certain range without revealing their real values.

- Second, we have proposed a privacy-preserving worker selection scheme for MCS based on probabilistic skyline query over sliding windows. The proposed scheme can continuously select reliable workers in terms of working experience, expiry time, and trustability without revealing sensitive information. More specifically, a probabilistic skyline approach is designed for a dynamic situation where workers may constantly arrive at/leave the platform. For protecting workers' privacy, we have designed an ElGamal-based encryption approach for securely outsourcing sensitive information and comparing workers' personal information.

- Third, we have proposed a privacy-preserving scheme for F-MCS applications, called FedSky. By extending the classic FedAvg algorithm, FedSky selects qualified workers based on CG-skyline technique and securely aggregates model updates for training the global model. Particularly, compared with FedAvg, our scheme considers the workers' dynamics and heterogeneous natures. It can significantly improve the efficiency of the model training process in F-MCS. Besides, we have designed a novel privacy-preserving data aggregation protocol based on the additive HE system. This protocol is exclusively designed for *cross-device* FL setting and requires no interactions between workers during the model training process.

- Fourth, we have proposed two privacy-preserving worker selection schemes in terms of both spatial-temporal similarity and spatial-temporal constraints in MCS services. We introduced the definition and calculation of spatial-temporal similarity and constraints between participating workers and the MCS task in terms of intersection area ratio and overlapping time duration. For the case of spatial-temporal similarity, we designed a privacy-preserving approach for protecting participants' sensitive information based on the modified Paillier encryption technique. For the case of spatial-temporal constraint, we proposed a privacy-preserving worker selection scheme based on the symmetric homomorphic encryption (SHE) approach, which can protect participants' spatial information from being disclosed while still enabling effective worker assignment.

10.2 Future Work

In the research area of privacy-preserving worker selection in MCS, there are still several potential and essential research problems to be explored in the future.

- Privacy-preserving Worker Selection in MCS over Trajectories on Road Network. Compared with the existing matching-based spatial-temporal worker selection problems, workers' trajectories contain more sensitive and personal information, which can be exploited by the attackers for daily routine analyses and daily behavior prediction. In our future work, we plan to design privacy-preserving worker selection schemes for future trajectories on road network, which can help the MCS platform select qualified workers based on workers' future or predictable trajectories without revealing any sensitive information in more practical scenarios.

- Privacy-preserving Worker Selection in MCS based on Workers' Mobility Prediction. In the existing MCS platforms, one of the workers' major concerns is the cost of uploading real-time data. In the future, we plan to consider a novel MCS scenario where a worker w_i can share his/her data with another worker w_j freely (e.g., under a Wi-Fi environment). Then worker w_j can upload the sensing data to the MCS platform with no charge. In this situation, the classic Markov model or semi-Markov model can be used to determine the contact probability between w_i and w_j. Furthermore, the certain privacy-preserving scheme should be designed to protect workers' real-time spatial-temporal information.

- Privacy-preserving Worker Selection in More Realistic Applications. In this thesis, we designed a framework called FedSky for the application scenario of F-MCS. The proposed scheme is evaluated on a real-world dataset, MNIST. In the future, we plan to evaluate the performance of FedSky in more realistic problems, such as real-time object detection and object classification. In such scenarios, the problems are more complex and the model should be optimized in a more comprehensive way to achieve satisfying results.

- Incentive Mechanism in Privacy-preserving Worker Selection for MCS applications. One of the fundamental problems in MCS is how to inspire workers' motivations and interests in participating in the MCS tasks. In the future work, we plan to focus on different approaches and techniques for building an effective incentive mechanism in MCS worker selection. Particularly, the designed incentive mechanism should select high-quality workers by accurately assessing workers' trustability and their bids. Furthermore, each worker's personal information, e.g., their asking price, should be properly protected in the entire worker selection process.

Bibliography

[1] Alaa Omran Almagrabi, Rashid Ali, Daniyal Alghazzawi, Abdullah Al-Barakati, and Tahir Khurshaid, *A reinforcement learning-based framework for crowdsourcing in massive health care internet of things*, Big data (2021).

[2] Miguel E Andrés, Nicolás E Bordenabe, Konstantinos Chatzikokolakis, and Catuscia Palamidessi, *Geo-indistinguishability: Differential privacy for location-based systems*, arXiv preprint arXiv:1212.1984 (2012).

[3] Yoshinori Aono, Takuya Hayashi, Lihua Wang, Shiho Moriai, et al., *Privacy-preserving deep learning via additively homomorphic encryption*, IEEE Transactions on Information Forensics and Security **13** (2017), no. 5, 1333–1345.

[4] Jaime Ballesteros, Bogdan Carbunar, Mahmudur Rahman, Naphtali Rishe, and SS Iyengar, *Towards safe cities: A mobile and social networking approach*, IEEE Transactions on Parallel and Distributed Systems **25** (2013), no. 9, 2451–2462.

[5] Ezedin Barka, Chaker Abdelaziz Kerrache, Nasreddine Lagraa, and Abderrahmane Lakas, *Behavior-aware uav-assisted crowd sensing technique for urban vehicular environments*, 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, 2018, pp. 1–7.

[6] Dan Boneh, *The decision diffie-hellman problem*, International Algorithmic Number Theory Symposium, Springer, 1998, pp. 48–63.

[7] Dario Bonino, Maria Teresa Delgado Alizo, Claudio Pastrone, and Maurizio Spirito, *Wasteapp: Smarter waste recycling for smart citizens*, 2016 International Multidisciplinary Conference on Computer and Energy Science (SpliTech), IEEE, 2016, pp. 1–6.

[8] Stephan Börzsönyi, Donald Kossmann, and Konrad Stocker, *The skyline operator*, Proceedings of the 17th International Conference on Data Engineering, April 2-6, 2001, Heidelberg, Germany, IEEE Computer Society, 2001, pp. 421–430.

[9] Djallel Eddine Boubiche, Muhammad Imran, Aneela Maqsood, and Muhammad Shoaib, *Mobile crowd sensing–taxonomy, applications, challenges, and solutions*, Computers in Human Behavior **101** (2019), 352–370.

[10] Emmanuel Bresson, Dario Catalano, and David Pointcheval, *A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications*, Advances in Cryptology - ASIACRYPT 2003, 9th International Conference on the Theory and Application of Cryptology and Information Security, Taipei, Taiwan, November 30 - December 4, 2003, Proceedings (Chi-Sung Laih, ed.), Lecture Notes in Computer Science, vol. 2894, Springer, 2003, pp. 37–54.

[11] Nirupama Bulusu, Chun Tung Chou, Salil Kanhere, Yifei Dong, Shitiz Sehgal, David Sullivan, and Lupco Blazeski, *Participatory sensing in commerce: Using mobile camera phones to track market price dispersion*, Proceedings of the international workshop on urban, community, and social applications of networked sensing systems (UrbanSense 2008), 2008, pp. 6–10.

[12] Andrea Capponi, Claudio Fiandrino, Burak Kantarci, Luca Foschini, Dzmitry Kliazovich, and Pascal Bouvry, *A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities*, IEEE communications surveys & tutorials **21** (2019), no. 3, 2419–2465.

[13] Cengn, *LIVING UP TO THE POTENTIAL OF IOT*, http://www.cengn.ca/iot-potential/, Accessed: 16-Feb-2020.

[14] Lijiang Chen, Bin Cui, and Hua Lu, *Constrained skyline query processing against distributed data sites*, IEEE Transactions on Knowledge and Data Engineering **23** (2010), no. 2, 204–217.

[15] Yueyue Chen, Pin Lv, Deke Guo, Tongqing Zhou, and Ming Xu, *Trajectory segment selection with limited budget in mobile crowd sensing*, Pervasive Mob. Comput. **40** (2017), 123–138.

[16] Peng Cheng, Xiang Lian, Zhao Chen, Rui Fu, Lei Chen, Jinsong Han, and Jizhong Zhao, *Reliable diversity-based spatial crowdsourcing by moving workers*, Proc. VLDB Endow. **8** (2015), no. 10, 1022–1033.

[17] Yohan Chon, Nicholas D Lane, Yunjong Kim, Feng Zhao, and Hojung Cha, *Understanding the coverage and scalability of place-centric crowdsensing*, Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing, 2013, pp. 3–12.

[18] Yohan Chon, Nicholas D Lane, Fan Li, Hojung Cha, and Feng Zhao, *Automatically characterizing places with opportunistic crowdsensing using smartphones*, Proceedings of the 2012 ACM conference on ubiquitous computing, 2012, pp. 481–490.

[19] Gao Cong, Christian S Jensen, and Dingming Wu, *Efficient retrieval of the top-k most relevant spatial web objects*, Proceedings of the VLDB Endowment **2** (2009), no. 1, 337–348.

[20] Vladimir Coric and Marco Gruteser, *Crowdsensing maps of on-street parking spaces*, 2013 IEEE International Conference on Distributed Computing in Sensor Systems, IEEE, 2013, pp. 115–122.

[21] Bin Cui, Hua Lu, Quanqing Xu, Lijiang Chen, Yafei Dai, and Yongluan Zhou, *Parallel distributed processing of constrained skyline queries by filtering*, 2008 IEEE 24th International Conference on Data Engineering, IEEE, 2008, pp. 546–555.

[22] Yue Cui, Liwei Deng, Yan Zhao, Bin Yao, Vincent W Zheng, and Kai Zheng, *Hidden poi ranking with spatial crowdsourcing*, Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 814–824.

[23] Tan Dan, Ruipeng Gao, and Hongbin Sun, *Sensing-gain constrained participant selection mechanism for mobile crowdsensing*, Personal and Ubiquitous Computing **17** (2020), no. 7, 1–15.

[24] Shideh Dashti, Jonathan D Bray, Jack Reilly, Steven Glaser, Alexandre Bayen, and Ervasti Mari, *Evaluating the reliability of phones as seismic monitoring instruments*, Earthquake Spectra **30** (2014), no. 2, 721–742.

[25] Ye Dong, Xiaojun Chen, Liyan Shen, and Dakui Wang, *Eastfly: Efficient and secure ternary federated learning*, Computers & Security **94** (2020), 101824.

[26] Yi F Dong, Salil Kanhere, Chun Tung Chou, and Nirupama Bulusu, *Automatic collection of fuel prices from a network of mobile cameras*, International Conference on Distributed Computing in Sensor Systems, Springer, 2008, pp. 140–156.

[27] Prabal Dutta, Paul M Aoki, Neil Kumar, Alan Mainwaring, Chris Myers, Wesley Willett, and Allison Woodruff, *Common sense: participatory urban sensing using a network of handheld air quality monitors*, Proceedings of the 7th ACM conference on embedded networked sensor systems, 2009, pp. 349–350.

[28] Newman Mark EJ., *Spread of epidemic disease on networks*, Physical review E **66** (2002), no. 1.

[29] Matthew Faulkner, Robert Clayton, Thomas Heaton, K Mani Chandy, Monica Kohler, Julian Bunn, Richard Guy, Annie Liu, Michael Olson, MingHei Cheng, et al., *Community sense and response systems: Your phone as quake detector*, Communications of the ACM **57** (2014), no. 7, 66–75.

[30] Niels Ferguson and Bruce Schneier, *Practical cryptography*, vol. 141, Wiley New York, 2006.

[31] Claudio Fiandrino, Andrea Capponi, Giuseppe Cacciatore, Dzmitry Kliazovich, Ulrich Sorger, Pascal Bouvry, Burak Kantarci, Fabrizio Granelli, and Stefano Giordano, *Crowdsensim: a simulation platform for mobile crowdsensing in realistic urban environments*, Ieee access **5** (2017), 3490–3503.

[32] Xiaowen Gong and Ness B Shroff, *Truthful mobile crowdsensing for strategic users with private data quality*, IEEE/ACM Transactions on Networking **27** (2019), no. 5, 1959–1972.

[33] Srinivasa Raghavendra Bhuvan Gummidi, Xike Xie, and Torben Bach Pedersen, *A survey of spatial crowdsourcing*, ACM Transactions on Database Systems (TODS) **44** (2019), no. 2, 1–46.

[34] Bin Guo, Huihui Chen, Zhiwen Yu, Wenqian Nan, Xing Xie, Daqing Zhang, and Xingshe Zhou, *Taskme: Toward a dynamic and quality-enhanced incentive mechanism for mobile crowd sensing*, International Journal of Human-Computer Studies **102** (2017), 14–26.

[35] Bin Guo, Yan Liu, Wenle Wu, Zhiwen Yu, and Qi Han, *Activecrowd: A framework for optimized multitask allocation in mobile crowdsensing systems*, IEEE Transactions on HMS **47** (2017), no. 3, 392–403.

[36] Weisi Guo and Siyi Wang, *Mobile crowd-sensing wireless activity with measured interference power*, IEEE Wirel. Commun. Lett. **2** (2013), no. 5, 539–542.

[37] Aymen Hamrouni, Hakim Ghazzai, Mounir Frikha, and Yehia Massoud, *A photo-based mobile crowdsourcing framework for event reporting*, 62nd IEEE International Midwest Symposium on Circuits and Systems, MWSCAS 2019, Dallas, TX, USA, August 4-7, 2019 (Hoi Lee and Randall L. Geiger, eds.), IEEE, 2019, pp. 198–202.

[38] Guangjie Han, Li Liu, Sammy Chan, Ruiyun Yu, and Yu Yang, *Hysense: A hybrid mobile crowdsensing framework for sensing opportunities compensation under dynamic coverage constraint*, IEEE Commun. Mag. **55** (2017), no. 3, 93–99.

[39] Meng Hao, Hongwei Li, Xizhao Luo, Guowen Xu, Haomiao Yang, and Sen Liu, *Efficient and privacy-enhanced federated learning for industrial artificial intelligence*, IEEE Transactions on Industrial Informatics **16** (2019), no. 10, 6532–6542.

[40] David Hasenfratz, Olga Saukh, Silvan Sturzenegger, Lothar Thiele, et al., *Participatory air pollution monitoring using smartphones*, Mobile Sensing **1** (2012), 1–5.

[41] Zongjian He, Jiannong Cao, and Xuefeng Liu, *High quality participant recruitment in vehicle-based crowdsourcing using predictable mobility*, 2015 IEEE Conference on Computer Communications (INFOCOM), IEEE, 2015, pp. 2542–2550.

[42] Qin Hu, Shengling Wang, Xiuzhen Cheng, Junshan Zhang, and Weifeng Lv, *Cost-efficient mobile crowdsensing with spatial-temporal awareness*, IEEE Trans. Mob. Comput. **20** (2021), no. 3, 928–938.

[43] Yang Ishigaki, Yoshinori Matsumoto, Ryo Ichimiya, and Kenji Tanaka, *Development of mobile radiation monitoring system utilizing smartphone and its field tests in fukushima*, IEEE Sensors Journal **13** (2013), no. 10, 3520–3526.

[44] Haiming Jin, Lu Su, Bolin Ding, Klara Nahrstedt, and Nikita Borisov, *Enabling privacy-preserving incentives for mobile crowd sensing systems*, 2016 IEEE 36th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2016, pp. 344–353.

[45] Haiming Jin, Lu Su, Houping Xiao, and Klara Nahrstedt, *Inception: Incentivizing privacy-preserving data aggregation for mobile crowd sensing systems*, Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2016, pp. 341–350.

[46] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al., *Advances and open problems in federated learning*, arXiv preprint arXiv:1912.04977 (2019).

[47] Merkouris Karaliopoulos, Iordanis Koutsopoulos, and Leonidas Spiliopoulos, *Optimal user choice engineering in mobile crowdsensing with bounded rational users*, IEEE INFOCOM 2019-IEEE Conference on Computer Communications, IEEE, 2019, pp. 1054–1062.

[48] Leyla Kazemi and Cyrus Shahabi, *Geocrowd: enabling query answering with spatial crowdsourcing*, Proceedings of the 20th international conference on advances in geographic information systems, ACM, 2012, pp. 189–198.

[49] Leyla Kazemi, Cyrus Shahabi, and Lei Chen, *Geotrucrowd: trustworthy query answering with spatial crowdsourcing*, Proceedings of the 21st acm sigspatial

international conference on advances in geographic information systems, ACM, 2013, pp. 314–323.

[50] Sunyoung Kim, Christine Robson, Thomas Zimmerman, Jeffrey Pierce, and Eben M Haber, *Creek watch: pairing usefulness and usability for successful citizen science*, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, 2011, pp. 2125–2134.

[51] Xujiao Kong, Peng Li, Tao Zhang, Meimei Tu, and Qin Liu, *A budget constraint incentive mechanism in spatial-temporal mobile crowdsensing*, 21st IEEE International Conference on High Performance Computing and Communications; 17th IEEE International Conference on Smart City; 5th IEEE International Conference on Data Science and Systems, HPCC/SmartCity/DSS 2019, Zhangjiajie, China, August 10-12, 2019 (Zheng Xiao, Laurence T. Yang, Pavan Balaji, Tao Li, Keqin Li, and Albert Y. Zomaya, eds.), IEEE, 2019, pp. 1726–1731.

[52] Thomas S Lendvay, Lee White, and Timothy Kowalewski, *Crowdsourcing to assess surgical skill*, JAMA surgery **150** (2015), no. 11, 1086–1087.

[53] Arjen K Lenstra and Eric R Verheul, *Selecting cryptographic key sizes*, Journal of cryptology **14** (2001), no. 4, 255–293.

[54] Beibei Li, Yuhao Wu, Jiarui Song, Rongxing Lu, Tao Li, and Liang Zhao, *Deepfed: Federated deep learning for intrusion detection in industrial cyber-physical systems*, IEEE Transactions on Industrial Informatics (2020).

[55] Meng Li, Liehuang Zhu, and Xiaodong Lin, *Privacy-preserving traffic monitoring with false report filtering via fog-assisted vehicular crowdsensing*, IEEE Transactions on Services Computing (2019).

[56] Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith, *Federated learning: Challenges, methods, and future directions*, IEEE Signal Processing Magazine **37** (2020), no. 3, 50–60.

[57] Xin Li and Xinglin Zhang, *Multi-task allocation under time constraints in mobile crowdsensing*, IEEE Trans. Mob. Comput. **20** (2021), no. 4, 1494–1510.

[58] Jian Lin, Dejun Yang, Ming Li, Jia Xu, and Guoliang Xue, *Frameworks for privacy-preserving mobile crowdsensing incentive mechanisms*, IEEE Transactions on Mobile Computing **17** (2017), no. 8, 1851–1864.

[59] Chi Harold Liu, Chengzhe Piao, and Jian Tang, *Energy-efficient uav crowdsensing with multiple charging stations by deep learning*, IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 199–208.

[60] Jinfei Liu, Li Xiong, Jian Pei, Jun Luo, Haoyu Zhang, and Wenhui Yu, *Group-based skyline for pareto optimal groups*, IEEE Transactions on Knowledge and Data Engineering (2019).

[61] Jinfei Liu, Haoyu Zhang, Li Xiong, Haoran Li, and Jun Luo, *Finding probabilistic k-skyline sets on uncertain data*, Proceedings of the 24th acm international on conference on information and knowledge management, ACM, 2015, pp. 1511–1520.

[62] Jinwei Liu, Haiying Shen, Husnu S Narman, Wingyan Chung, and Zongfang Lin, *A survey of mobile crowdsensing techniques: A critical component for the internet of things*, ACM Transactions on CSP **2** (2018), no. 3, 18.

[63] Yan Liu, Bin Guo, Yang Wang, Wenle Wu, Zhiwen Yu, and Daqing Zhang, *Taskme: Multi-task allocation in mobile crowd sensing*, Proceedings of the 2016

ACM International Joint Conference on Pervasive and Ubiquitous Computing, ACM, 2016, pp. 403–414.

[64] Rongxing Lu, Xiaohui Liang, Xu Li, Xiaodong Lin, and Xuemin Shen, *Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications*, IEEE Transactions on Parallel and Distributed Systems **23** (2012), no. 9, 1621–1631.

[65] Rongxing Lu, Xiaodong Lin, Haojin Zhu, P-H Ho, and Xuemin Shen, *Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications*, IEEE INFOCOM 2008-The 27th Conference on Computer Communications, IEEE, 2008, pp. 1229–1237.

[66] Yu Lu, Shili Xiang, Wei Wu, and Huayu Wu, *A queue analytics system for taxi service using mobile crowd sensing*, Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2015 ACM International Symposium on Wearable Computers, UbiComp/ISWC Adjunct 2015, Osaka, Japan, September 7-11, 2015 (Kenji Mase, Marc Langheinrich, Daniel Gatica-Perez, Hans Gellersen, Tanzeem Choudhury, and Koji Yatani, eds.), ACM, 2015, pp. 121–124.

[67] Yunlong Lu, Xiaohong Huang, Yueyue Dai, Sabita Maharjan, and Yan Zhang, *Blockchain and federated learning for privacy-preserved data sharing in industrial iot*, IEEE Transactions on Industrial Informatics **16** (2019), no. 6, 4177–4186.

[68] Hassan Mahdikhani, Rongxing Lu, Yandong Zheng, Jun Shao, and Ali A Ghorbani, *Achieving o ($log^3 n$) communication-efficient privacy-preserving range query in fog-based iot*, IEEE Internet of Things Journal **7** (2020), no. 6, 5220–5232.

[69] Kalikinkar Mandal and Guang Gong, *Privfl: Practical privacy-preserving federated regressions on high-dimensional data over mobile networks*, Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop, 2019, pp. 57–68.

[70] H. Brendan McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas, *Federated learning of deep networks using model averaging*, CoRR **abs/1602.05629** (2016).

[71] Muntazir Mehdi, Guido Mühlmeier, Kushal Agrawal, Rüdiger Pryss, Manfred Reichert, and Franz J. Hauck, *Referenceable mobile crowdsensing architecture: A healthcare use case*, The 15th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2018) / The 13th International Conference on Future Networks and Communications (FNC-2018) / Affiliated Workshops, Gran Canaria, Spain, August 13-15, 2018 (Ansar-Ul-Haque Yasar and Elhadi M. Shakshuki, eds.), Procedia Computer Science, vol. 134, Elsevier, 2018, pp. 445–451.

[72] Qipei Mei, Mustafa Gül, and Nima Shirzad-Ghaleroudkhani, *Towards smart cities: crowdsensing-based monitoring of transportation infrastructure using in-traffic vehicles*, Journal of Civil Structural Health Monitoring **10** (2020), 653–665.

[73] Mohamed Musthag and Deepak Ganesan, *Labor dynamics in a mobile micro-task market*, 2013 ACM SIGCHI Conference on Human Factors in Computing Systems, CHI '13, Paris, France, April 27 - May 2, 2013 (Wendy E. Mackay, Stephen A. Brewster, and Susanne Bødker, eds.), ACM, 2013, pp. 641–650.

[74] Luong Vuong Nguyen, Jason J. Jung, and Myunggwon Hwang, *Ourplaces: Cross-cultural crowdsourcing platform for location recommendation services*, ISPRS Int. J. Geo Inf. **9** (2020), no. 12, 711.

[75] Jianbing Ni, Kuan Zhang, Xiaodong Lin, Qi Xia, and Xuemin Sherman Shen, *Privacy-preserving mobile crowdsensing for located-based applications*, ICC'17, IEEE, 2017, pp. 1–6.

[76] Wangze Ni, Peng Cheng, Lei Chen, and Xuemin Lin, *Task allocation in dependency-aware spatial crowdsourcing*, 36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020, IEEE, 2020, pp. 985–996.

[77] Takayuki Nishio and Ryo Yonetani, *Client selection for federated learning with heterogeneous resources in mobile edge*, ICC 2019-2019 IEEE International Conference on Communications (ICC), IEEE, 2019, pp. 1–7.

[78] Xin Niu, Hongyu Huang, and Yantao Li, *A real-time data collection mechanism with trajectory privacy in mobile crowd-sensing*, IEEE Commun. Lett. **24** (2020), no. 10, 2114–2118.

[79] Aart Overeem, JC R. Robinson, Hidde Leijnse, Gert-Jan Steeneveld, BK P. Horn, and Remko Uijlenhoet, *Crowdsourcing urban air temperatures from smartphone battery temperatures*, Geophysical Research Letters **40** (2013), no. 15, 4081–4085.

[80] Christof Paar and Jan Pelzl, *Understanding cryptography: a textbook for students and practitioners*, Springer Science & Business Media, 2009.

[81] Anita Pal and Madhumangal Pal, *Interval tree and its applications*, Advanced Modeling and Optimization **11** (2009), no. 3, 211–224.

[82] Victor Pankratius, Frank Lind, Anthea Coster, Philip Erickson, and Joshua Semeter, *Mobile crowd sensing in space weather monitoring: the mahali project*, IEEE communications magazine **52** (2014), no. 8, 22–28.

[83] Dimitris Papadias, Yufei Tao, Greg Fu, and Bernhard Seeger, *An optimal and progressive algorithm for skyline queries*, Proceedings of the 2003 ACM SIGMOD international conference on Management of data, 2003, pp. 467–478.

[84] Jian Pei, Bin Jiang, Xuemin Lin, and Yidong Yuan, *Probabilistic skylines on uncertain data*, Proceedings of the 33rd international conference on Very large data bases, VLDB Endowment, 2007, pp. 15–26.

[85] A Pentland, N Eagle, and D Lazer, *Inferring social network structure using mobile phone data*, Proceedings of the National Academy of Sciences (PNAS) **106** (2009), no. 36, 15274–15278.

[86] Dieter Pfoser and Christian S. Jensen, *Capturing the uncertainty of moving-object representations*, Advances in Spatial Databases, 6th International Symposium, SSD'99, Hong Kong, China, July 20-23, 1999, Proceedings (Ralf Hartmut Güting, Dimitris Papadias, and Frederick H. Lochovsky, eds.), Lecture Notes in Computer Science, vol. 1651, Springer, 1999, pp. 111–132.

[87] Rüdiger Pryss, Manfred Reichert, Berthold Langguth, and Winfried Schlee, *Mobile crowd sensing services for tinnitus assessment, therapy, and research*, 2015 IEEE International Conference on Mobile Services, IEEE, 2015, pp. 352–359.

[88] Md Abdur Rahman and M Shamim Hossain, *A location-based mobile crowdsensing framework supporting a massive ad hoc social network environment*, IEEE Communications Magazine **55** (2017), no. 3, 76–85.

[89] Jack Reilly, Shideh Dashti, Mari Ervasti, Jonathan D Bray, Steven D Glaser, and Alexandre M Bayen, *Mobile phones as seismologic sensors: Automating data extraction for the ishake system*, IEEE Transactions on Automation Science and Engineering **10** (2013), no. 2, 242–251.

[90] Ju Ren, Yaoxue Zhang, Kuan Zhang, and Xuemin Sherman Shen, *Sacrm: Social aware crowdsourcing with reputation management in mobile sensing*, Computer Communications **65** (2015), 55–65.

[91] Samad Saadatmand and Salil S Kanhere, *Mra: A modified reverse auction based framework for incentive mechanisms in mobile crowdsensing systems*, Computer Communications **145** (2019), 137–145.

[92] Hanan Samet, *Foundations of multidimensional and metric data structures*, Morgan Kaufmann series in data management systems, Academic Press, 2006.

[93] William Schneble and Geethapriya Thamilarasu, *Attack detection using federated learning in medical cyber-physical systems*, 2019 28th International Conference on Computer Communication and Networks, ICCCN, 2019, pp. 1–8.

[94] Shitiz Sehgal, Salil S Kanhere, and Chun Tung Chou, *Mobishop: Using mobile phones for sharing consumer pricing information*, Demo Session of the Intl. Conference on Distributed Computing in Sensor Systems, vol. 13, 2008.

[95] Kewei Sha, Guoxing Zhan, Weisong Shi, Mark Lumley, Clairy Wiholm, and Bengt Arnetz, *Spa: a smart phone assisted chronic illness self-management system with participatory sensing*, Proceedings of the 2nd International Workshop on Systems and Networking Support for Health Care and Assisted Living Environments, 2008, pp. 1–3.

[96] Shuo Shang, Lisi Chen, Zhewei Wei, Christian Søndergaard Jensen, Kai Zheng, and Panos Kalnis, *Trajectory similarity join in spatial networks*, Proceedings of the VLDB Endowment **10** (2017), no. 11.

[97] M Sim, J Bigham, H Kautz, and M Halterman, *Crowdsourcing medical expertise in near real time*, Journal of Hospital Medicine **9** (2014), no. 7, 451–456.

[98] Vijay Sivaraman, James Carrapetta, Ke Hu, and Blanca Gallego Luxan, *Hazewatch: A participatory sensor system for monitoring air pollution in sydney*, 38th Annual IEEE Conference on Local Computer Networks-Workshops, IEEE, 2013, pp. 56–64.

[99] Dezhi Sun, Ke Xu, Hao Cheng, Yuanyuan Zhang, Tianshu Song, Rui Liu, and Yi Xu, *Online delivery route recommendation in spatial crowdsourcing*, World Wide Web **22** (2019), no. 5, 2083–2104.

[100] Peng Sun, Zhibo Wang, Yunhe Feng, Liantao Wu, Yanjun Li, Hairong Qi, and Zhi Wang, *Towards personalized privacy-preserving incentive for truth discovery in crowdsourced binary-choice question answering*, IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1133–1142.

[101] Sunita Tiwari and Saroj Kaushik, *Information enrichment for tourist spot recommender system using location aware crowdsourcing*, 2014 IEEE 15th International Conference on Mobile Data Management, vol. 2, IEEE, 2014, pp. 11–14.

[102] Hien To, Cyrus Shahabi, and Li Xiong, *Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server*, 34th IEEE International Conference on Data Engineering, ICDE 2018, Paris, France, April 16-19, 2018, IEEE Computer Society, 2018, pp. 833–844.

[103] Yongxin Tong, Lei Chen, and Cyrus Shahabi, *Spatial crowdsourcing: Challenges, techniques, and applications*, Proceedings of the VLDB Endowment **10** (2017), no. 12, 1988–1991.

[104] Yongxin Tong, Jieying She, Bolin Ding, Libin Wang, and Lei Chen, *Online mobile micro-task allocation in spatial crowdsourcing*, 2016 IEEE 32Nd international conference on data engineering (ICDE), IEEE, 2016, pp. 49–60.

[105] Goce Trajcevski, Alok Choudhary, Ouri Wolfson, Li Ye, and Gang Li, *Uncertain range queries for necklaces*, 2010 Eleventh International Conference on Mobile Data Management, IEEE, 2010, pp. 199–208.

[106] Hao Wang, Zakhary Kaplan, Di Niu, and Baochun Li, *Optimizing federated learning on non-iid data with reinforcement learning*, IEEE INFOCOM 2020-IEEE Conference on Computer Communications, IEEE, 2020, pp. 1698–1707.

[107] Leye Wang, Dingqi Yang, Xiao Han, Tianben Wang, Daqing Zhang, and Xiaojuan Ma, *Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation*, Proceedings of the 26th International Conference on World Wide Web, International World Wide Web Conferences Steering Committee, 2017, pp. 627–636.

[108] Xiaojie Wang, Zhaolong Ning, Xiping Hu, Edith C-H Ngai, Lei Wang, Bin Hu, and Ricky YK Kwok, *A city-wide real-time traffic management system: Enabling crowdsensing in social internet of vehicles*, IEEE Communications Magazine **56** (2018), no. 9, 19–25.

[109] Xiong Wang, Jinbei Zhang, Xiaohua Tian, Xiaoying Gan, Yunfeng Guan, and Xinbing Wang, *Crowdsensing-based consensus incident report for road traffic acquisition*, IEEE transactions on intelligent transportation systems **19** (2017), no. 8, 2536–2547.

[110] Xiumin Wang, Weiwei Wu, and Deyu Qi, *Mobility-aware participant recruitment for vehicle-based mobile crowdsensing*, IEEE Trans. Veh. Technol. **67** (2018), no. 5, 4415–4426.

[111] Zhibo Wang, Jiahui Hu, Ruizhao Lv, Jian Wei, Qian Wang, Dejun Yang, and Hairong Qi, *Personalized privacy-preserving task allocation for mobile crowdsensing*, IEEE Transactions on Mobile Computing (2018).

[112] Zhibo Wang, Xiaoyi Pang, Jiahui Hu, Wenxin Liu, Qian Wang, Yanjun Li, and Honglong Chen, *When mobile crowdsensing meets privacy*, IEEE Communications Magazine **57** (2019), no. 9, 72–78.

[113] Wikipedia, *WAZE, a GPS navigation software app*, https://en.wikipedia.org/wiki/Waze, Accessed: 2019-09-08.

[114] Yibo Wu, Yi Wang, and Guohong Cao, *Photo crowdsourcing for area coverage in resource constrained environments*, 2017 IEEE Conference on Computer Communications, INFOCOM 2017, Atlanta, GA, USA, May 1-4, 2017, IEEE, 2017, pp. 1–9.

[115] Yibo Wu, Yi Wang, Wenjie Hu, Xiaomei Zhang, and Guohong Cao, *Resource-aware photo crowdsourcing through disruption tolerant networks*, 36th IEEE International Conference on Distributed Computing Systems, ICDCS 2016, Nara, Japan, June 27-30, 2016, IEEE Computer Society, 2016, pp. 374–383.

[116] Yue Wu, Fan Li, Liran Ma, Yadong Xie, Ting Li, and Yu Wang, *A context-aware multiarmed bandit incentive mechanism for mobile crowd sensing systems*, IEEE Internet of Things Journal **6** (2019), no. 5, 7648–7658.

[117] Han Xiao, Kashif Rasul, and Roland Vollgraf, *Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms*, arXiv preprint arXiv:1708.07747 (2017).

[118] Haoyi Xiong, Daqing Zhang, Leye Wang, J Paul Gibson, and Jie Zhu, *Eemc: Enabling energy-efficient mobile crowdsensing with anonymous participants*, ACM Transactions on Intelligent Systems and Technology (TIST) **6** (2015), no. 3, 39.

[119] Guowen Xu, Hongwei Li, and Rongxing Lu, *Practical and privacy-aware truth discovery in mobile crowd sensing systems*, Proceedings of the 2018 ACM

SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018 (David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, eds.), ACM, 2018, pp. 2312–2314.

[120] Qiang Xu and Rong Zheng, *Mobibee: A mobile treasure hunt game for location-dependent fingerprint collection*, Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, 2016, pp. 1472–1477.

[121] Qiang Xu, Rong Zheng, and Ezzeldin Tahoun, *Detecting location fraud in indoor mobile crowdsensing*, Proceedings of the First ACM Workshop on Mobile Crowdsensing Systems and Applications, 2017, pp. 44–49.

[122] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, and Heiko Ludwig, *Hybridalpha: An efficient approach for privacy-preserving federated learning*, Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, 2019, pp. 13–23.

[123] Yifan Xu, Jun Tao, Yang Gao, and Li Zeng, *Location-aware worker selection for mobile opportunistic crowdsensing in vanets*, GLOBECOM'17, IEEE, 2017, pp. 1–6.

[124] Qiang Yang, Yang Liu, Yong Cheng, Yan Kang, Tianjian Chen, and Han Yu, *Federated learning*, Synthesis Lectures on Artificial Intelligence and Machine Learning **13** (2019), no. 3, 1–207.

[125] Jiapeng Yu, Mingjun Xiao, Guoju Gao, and Chang Hu, *Minimum cost spatial-temporal task allocation in mobile crowdsensing*, Wireless Algorithms, Systems, and Applications - 11th International Conference, WASA 2016, Bozeman, MT, USA, August 8-10, 2016. Proceedings (Qing Yang, Wei Yu, and Yacine Challal,

eds.), Lecture Notes in Computer Science, vol. 9798, Springer, 2016, pp. 262–271.

[126] Bo Zhang, Chi Harold Liu, Jian Tang, Zhiyuan Xu, Jian Ma, and Wendong Wang, *Learning-based energy-efficient data collection by unmanned vehicles in smart cities*, IEEE Transactions on Industrial Informatics **14** (2017), no. 4, 1666–1676.

[127] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu, *Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning*, 2020 Annual Technical Conference (USENIX, ATC), 2020, pp. 493–506.

[128] Chuan Zhang, Liehuang Zhu, Chang Xu, Xiaojiang Du, and Mohsen Guizani, *A privacy-preserving traffic monitoring scheme via vehicular crowdsourcing*, Sensors **19** (2019), no. 6, 1274.

[129] Daqing Zhang, Lin Sun, Bin Li, Chao Chen, Gang Pan, Shijian Li, and Zhaohui Wu, *Understanding taxi service strategies from taxi GPS traces*, IEEE Trans. Intell. Transp. Syst. **16** (2015), no. 1, 123–135.

[130] Daqing Zhang, Leye Wang, Haoyi Xiong, and Bin Guo, *4w1h in mobile crowd sensing*, IEEE Communications Magazine **52** (2014), no. 8, 42–48.

[131] Maotian Zhang, Panlong Yang, Chang Tian, Shaojie Tang, Xiaofeng Gao, Baowei Wang, and Fu Xiao, *Quality-aware sensing coverage in budget-constrained mobile crowdsensing networks*, IEEE Trans. Veh. Technol. **65** (2016), no. 9, 7698–7707.

[132] Songnian Zhang, Suprio Ray, Rongxing Lu, Yandong Zheng, and Jun Shao, *Preserving location privacy for outsourced most-frequent item query in mobile crowdsensing*, IEEE Internet Things J. **8** (2021), no. 11, 9139–9150.

[133] Wenjie Zhang, Xuemin Lin, Ying Zhang, Wei Wang, and Jeffrey Xu Yu, *Probabilistic skyline operator over sliding windows*, 2009 IEEE 25th International Conference on Data Engineering, IEEE, 2009, pp. 1060–1071.

[134] Xichen Zhang and Ali A Ghorbani, *Human factors in cybersecurity: Issues and challenges in big data*, Security, Privacy, and Forensics Issues in Big Data, IGI Global, 2020, pp. 66–96.

[135] Xichen Zhang, Rongxing Lu, Jun Shao, Hui Zhu, and Ali A Ghorbani, *Achieve secure and efficient skyline computation for worker selection in mobile crowdsensing*, 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), IEEE, 2019, pp. 1–6.

[136] Xichen Zhang, Rongxing Lu, Jun Shao, Hui Zhu, and Ali A. Ghorbani, *Secure and efficient probabilistic skyline computation for worker selection in MCS*, IEEE Internet Things J. **7** (2020), no. 12, 11524–11535.

[137] Xichen Zhang, Rongxing Lu, Jun Shao, Hui Zhu, and Ali A. Ghorbani, *Continuous probabilistic skyline query for secure worker selection in mobile crowdsensing*, IEEE Internet Things J. **8** (2021), no. 14, 11758–11772.

[138] Xichen Zhang, Suprio Ray, Farzaneh Shoeleh, and Rongxing Lu, *Efficient contact similarity query over uncertain trajectories*, Proceedings of the 24th International Conference on Extending Database Technology, EDBT 2021, Nicosia, Cyprus, March 23 - 26, 2021, OpenProceedings.org, pp. 403–408.

[139] Bowen Zhao, Shaohua Tang, Ximeng Liu, and Xinglin Zhang, *Pace: Privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing*, IEEE Transactions on Mobile Computing **20** (2020), no. 5, 1924–1939.

[140] Qingchuan Zhao, Chaoshun Zuo, Giancarlo Pellegrino, and Zhiqiang Lin, *Geo-locating drivers: A study of sensitive data leakage in ride-hailing services*, 26th

Annual Network and Distributed System Security Symposium (NDSS), 2019, San Diego, California, USA, February 24-27, 2019, The Internet Society, 2019.

[141] Xuefeng Zhao, Niannian Wang, Ruicong Han, Botao Xie, Yan Yu, Mingchu Li, and Jinping Ou, *Urban infrastructure safety system based on mobile crowdsensing*, International journal of disaster risk reduction **27** (2018), 427–438.

[142] Yan Zhao, Yang Li, Yu Wang, Han Su, and Kai Zheng, *Destination-aware task assignment in spatial crowdsourcing*, Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, 2017, pp. 297–306.

[143] Yan Zhao, Kai Zheng, Yue Cui, Han Su, Feida Zhu, and Xiaofang Zhou, *Predictive task assignment in spatial crowdsourcing: a data-driven approach*, 2020 IEEE 36th International Conference on Data Engineering (ICDE), IEEE, 2020, pp. 13–24.

[144] Yandong Zheng, Rongxing Lu, Yunguo Guan, Jun Shao, and Hui Zhu, *Efficient and privacy-preserving similarity range query over encrypted time series data*, IEEE Transactions on Dependable and Secure Computing (2021).

[145] Yandong Zheng, Rongxing Lu, Beibei Li, Jun Shao, Haomiao Yang, and Kim-Kwang Raymond Choo, *Efficient privacy-preserving data merging and skyline computation over multi-source encrypted data*, Information Sciences **498** (2019), 91–105.

[146] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang, *Urban computing: concepts, methodologies, and applications*, ACM Transactions on Intelligent Systems and Technology (TIST) **5** (2014), no. 3, 1–55.

[147] Hui Zhu, Qing Wei, Xiaopeng Yang, Rongxing Lu, and Hui Li, *Efficient and privacy-preserving online fingerprint authentication scheme over outsourced data*, IEEE Transactions on Cloud Computing (2018).

[148] Xiaoyu Zhu, Yueyi Luo, Anfeng Liu, Wenjuan Tang, and Md. Zakirul Alam Bhuiyan, *A deep learning-based mobile crowdsensing scheme by predicting vehicle mobility*, IEEE Trans. Intell. Transp. Syst. **22** (2021), no. 7, 4648–4659.

Appendix A

The Concepts and Definitions about Contact Similarity

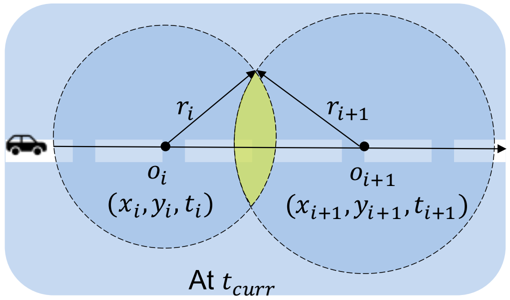
A.1 Trajectories, Beads, and Necklaces

In this section, we introduce the definitions of trajectories, beads, and necklaces. A trajectory is defined as follows.

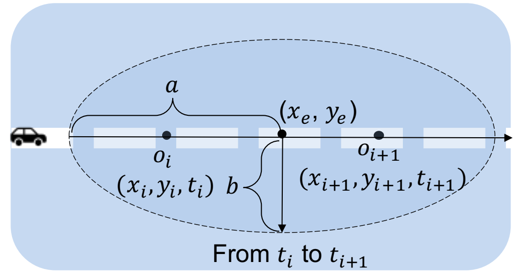
Definition A.1.1. *A **trajectory** T of a moving object (MO) is defined as finite, time-ordered observations $T = \{o_1, o_2, ..., o_n\}$, where $o_i = (x_i, y_i, t_i)$ for $i \in [1, n]$, with (x_i, y_i) being a sample point in Euclidean space, and t_i being a timestamp.*

The possible locations of a MO in-between two observations can be defined by a bead (or ellipse) [105] as follows.

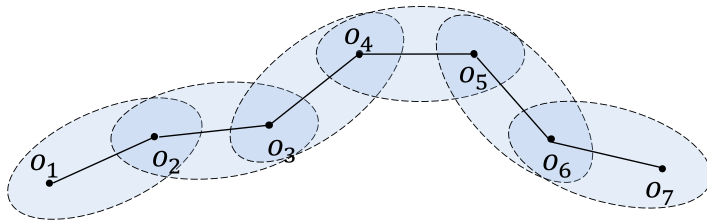
Definition A.1.2. *Let v_{max} denote the maximum speed that an object can take between o_i and o_{i+1}. A **bead** $B_i = \{(x_i, y_i, t_i), (x_{i+1}, y_{i+1}, t_{i+1})\}$ is defined as all*



(a) The possible locations of the MO at time $t_{curr}(t_i \leq t_{curr} \leq t_{i+1})$



(b) All the possible locations (the bead $\mathtt{B}_i$) of the MO between t_i and t_{i+1}.



(c) An example of necklace for a given uncertain trajectory with 7 observations.

Figure A.1: An example of bead $\mathtt{B}_i$ given the information of o_i, o_{i+1}, and v_{max}.

points (x, y, t) that satisfy the following constraints:

$$\begin{cases} t_i \leq t \leq t_{i+1} \\ (x - x_i)^2 + (y - y_i)^2 \leq (t - t_i)^2 \times v_{max}^2 \\ (x - x_{i+1})^2 + (y - y_{i+1})^2 \leq (t_{i+1} - t)^2 \times v_{max}^2. \end{cases} \tag{A.1}$$

From Fig. A.1 (a), we can see that from time t_i to t_{curr} ($t_{curr} > t_i$), the MO's possible travel area is a circle with (x_i, y_i) being the center and $r_i = (t_{curr} - t_i) \times v_{max}$ being the radius. Similarly, from time t_{curr} to t_{i+1} ($t_{curr} < t_{i+1}$), the possible locations are included in a circle centered at (x_{i+1}, y_{i+1}) with radius $r_{i+1} = (t_{i+1} - t_{curr}) \times v_{max}$. So, the overlapped area in Fig. A.1 (a) of the two circles includes MO's possible locations at current time t_{curr}. Based on [86], all the possible locations from time t_i to t_{i+1} form an ellipse (or bead) with foci at (x_i, y_i) and (x_{i+1}, y_{i+1}) (see in Fig. A.1

(b)). According to [105], the equation of the bead $\mathtt{B}$ is:

$$(x - x_e)^2/a^2 + (y - y_e)^2/b^2 = 1, \tag{A.2}$$

where (x_e, y_e) is the center of the ellipse; a and b represents the semi-major axis and semi-minor axis of the ellipse, as follows.

$$\begin{aligned} x_e &= \frac{x_i + x_{i+1}}{2}, \quad y_e = \frac{y_i + y_{i+1}}{2}, \quad a = \frac{v_{max} \times (t_{i+1} - t_i)}{2} \\ b &= \frac{\sqrt{v_{max}^2 \times (t_{i+1} - t_i)^2 - (x_{i+1} - x_i)^2 - (y_{i+1} - y_i)^2}}{2}. \end{aligned} \tag{A.3}$$

Definition A.1.3. *A trajectory $\mathtt{T} = \{o_1, o_2, ..., o_n\}$ can be represented as a sequence of beads, which is called a* ***necklace*** *and denoted as $\mathtt{T}_n = \{\mathtt{B}_1, \mathtt{B}_2, ..., \mathtt{B}_{n-1}\}$ (see an example in Fig. A.1 (c)), such that o_i and o_{i+1} form the bead $\mathtt{B}_i$ for $i \in [1, n-1]$.*

A.2 Contact Similarity

In this section, we introduce the essential definitions for contact similarity.

Definition A.2.1. *A* ***virtual landmark*** *$\mathtt{P}$ is a place where two MOs can contact with each other. $\mathtt{P}$ can be either meaningful locations (e.g., a shopping district or a park) or non-meaningful locations (e.g., a crossroad). For simplicity sake, any $\mathtt{P}$ in this work is considered as a circle that is centered at $O_c = (x_c, y_c)$ with a given radius r_c. Hence, the equation of $\mathtt{P}$ is $(x - x_c)^2 + (y - y_c)^2 = r_c^2$.*

Definition A.2.2. *For a MO m_1, he/she stayed at $\mathtt{P}$ for a certain period. For another MO m_2, his/her necklace is $\mathtt{T}_n = \{\mathtt{B}_1, \mathtt{B}_2, ..., \mathtt{B}_{n-1}\}$. If $\mathtt{P}$ spatially overlaps with one bead $\mathtt{B}_i \in \mathtt{T}_n$ for $i \in [1, n-1]$ (denoted as $\mathtt{P} \cap_S \mathtt{B}_i \neq \emptyset$), then the* ***intersection area ratio*** *$\mathtt{R}_{int}$ is*

$$\mathtt{R}_{int} = \mathtt{Area}_{int} / \mathtt{Area}_P, \tag{A.4}$$

where $\texttt{Area}_{int}$ *is the intersection area between the* $\texttt{P}$ *and* $\texttt{B}_i$*, and* $\texttt{Area}_P$ *is the area of* $\texttt{P}$*, i.e.,* $\texttt{Area}_P = \pi \cdot r_c^2$.

Fig. A.2 shows different scenarios that $\texttt{P}$ (the circle) overlaps with a bead $\texttt{B}_i$ (the ellipse), with two, three, and four intersection points, respectively. Then, we define the possible longest contact time duration between MOs as follows.

Definition A.2.3. *For MO* m_1*, he/she stayed at* $\texttt{P}$ *from time* t^s *to* t^e*. For another MO* m_2*, his/her necklace* $\texttt{T}_n = \{\texttt{B}_1, \texttt{B}_2, ..., \texttt{B}_{n-1}\}$*. Moreover, if one of the beads* $\texttt{B}_i \in \texttt{T}_n$ *(i.e.,* $i \in [1, n-1]$*) temporally overlaps with* $\texttt{P}$ *(denoted as* $\texttt{P} \cap_T \texttt{B}_i \neq \emptyset$*), then the* ***possible longest time duration*** $\tau(\texttt{P}, \texttt{B}_i)$ *between* m_1 *and* m_2 *can be calculated as*

$$\tau(\texttt{P}, \texttt{B}_i) = \min\{t^e, \bar{t}^e\} - \max\{t^s, \bar{t}^s\}, \tag{A.5}$$

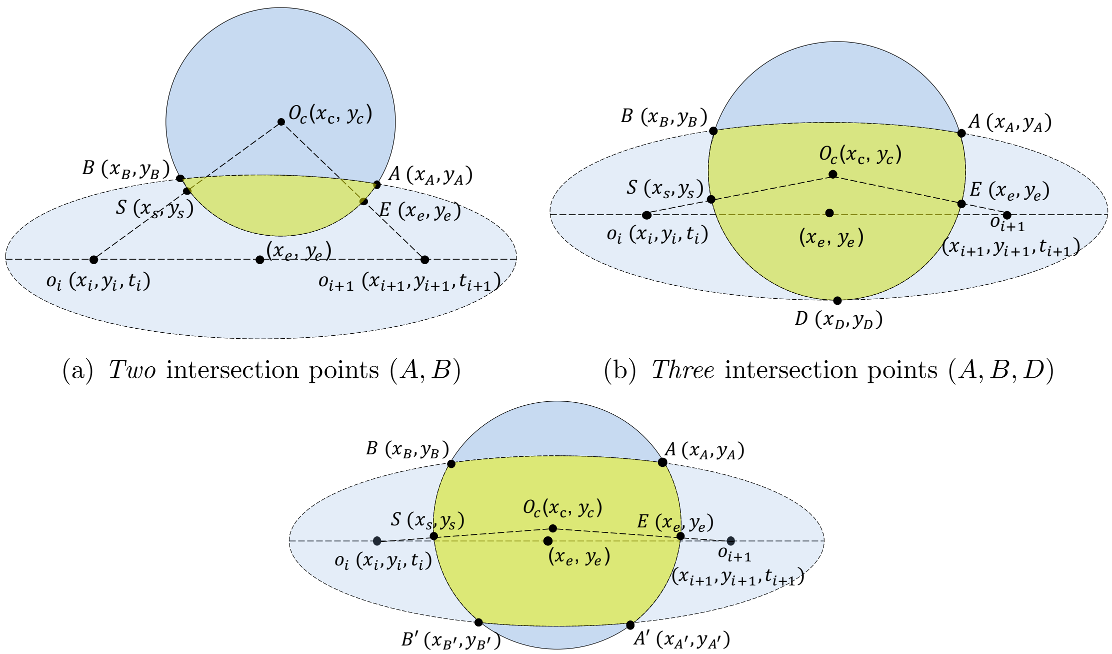
where $\bar{t}^s$ *means the possible earliest time for* m_2 *to arrive at* $\texttt{P}$*, and* $\bar{t}^e$ *means the possible latest time for* m_2 *to leave* $\texttt{P}$*.*

Example A.2.1. *If* m_1 *stayed at* $\texttt{P}$ *from* t^s *= 10:00 am and* t^e *= 10:50 am. For* m_2*,* $\bar{t}^s$ *= 10:05 am and* $\bar{t}^e$ *= 10:35 am, then* $\tau(\texttt{P}, \texttt{B}_i) = min\{$*10:50, 10:35*$\}$ *- max*$\{$*10:00, 10:05*$\}$ *= 30 min.*

Next, inspired by [28], we formally propose the definition of contact similarity as below.

Definition A.2.4. *For* m_1*, he/she stayed at* $\texttt{P}$ *from time* t^s *to* t^e*. For* m_2*, his/her necklace is* $\texttt{T}_n = \{\texttt{B}_1, \texttt{B}_2, ..., \texttt{B}_{n-1}\}$*. If* $\exists \texttt{B}_i \in \texttt{T}_n$ *such that* $\texttt{P} \cap_S \texttt{B}_i \neq \emptyset$ *and* $\texttt{P} \cap_T \texttt{B}_i \neq \emptyset$*, then* ***the contact similarity at bead level*** $Sim_{cont}^{\texttt{P}\cap\texttt{B}_i}$ *between* m_1 *and* m_2 *at* $\texttt{P}$ *in terms of* $\texttt{B}_i$ *can be calculated as:*

$$Sim_{cont}^{\texttt{P}\cap\texttt{B}_i} = 1 - (1 - \texttt{R}_{int})^{\tau(\texttt{P}, \texttt{B}_i)} \tag{A.6}$$



(a) *Two* intersection points (A, B)

(b) *Three* intersection points (A, B, D)

(c) *Four* intersection points (A, B, A', B')

Figure A.2: Examples of overlap between a virtual landmark and a bead with different intersection points

We assume that the intersection area ratio $\mathtt{R}_{int}$ indicates the possible opportunities that m_1 may be in contact with m_2 at $\mathtt{P}$ per unit time. Therefore, $(1 - \mathtt{R}_{int})^{\tau(\mathtt{P},\mathtt{B}_i)}$ denotes the probability that m_1 and m_2 have no effective contact during the time period $\tau(\mathtt{P}, \mathtt{B}_i)$. Consequently, $Sim_{cont}^{\mathtt{P}\cap\mathtt{B}_i}(m_1, m_2) = 1-(1-\mathtt{R}_{int})^{\tau(\mathtt{P},\mathtt{B}_i)}$. A larger value of $Sim_{cont}^{\mathtt{P}\cap\mathtt{B}_i}(m_1, m_2)$ indicates a higher probability that an effective contact may happen between m_1 and m_2 at $\mathtt{P}$.

Definition A.2.5. *Let $\mathbb{B}$ denote a set of beads in $\mathtt{T}_n$, such that for $\forall B_i \in \mathbb{B}$, $P \cap_S B_i \neq \emptyset$ and $P \cap_T B_i \neq \emptyset$. Then,* ***the contact similarity at trajectory level*** *$Sim_{cont}^{P\cap T}$ can be calculated as:*

$$Sim_{cont}^{P\cap T} = \frac{\sum_{B_i \in \mathbb{B}} Sim_{cont}^{P\cap B_i}}{|\mathbb{B}|}. \tag{A.7}$$

Here, $|\mathbb{B}|$ denotes the number of beads in $\mathbb{B}$.

A.3 Calculation of $\texttt{Area}_{int}$ and $\tau(\texttt{P}, \texttt{B}_i)$

In this section, we discuss how to calculate $\texttt{Area}_{int}$ and $\tau(\texttt{P}, \texttt{B}_i)$, followed by the workflow for *CSQ*.

Given the $\texttt{P}$ with equation $(x - x_c)^2 + (y - y_c)^2 = r_c^2$ and the $\texttt{B}_i$ with equation $(x - x_e)^2/a^2 + (y - y_e)^2/b^2 = 1$, it is straightforward to calculate $\texttt{Area}_{int}$ as follows.

• There are two or three intersection points between the virtual landmark $\texttt{P}$ and the bead $\texttt{B}_i$ (see Fig. A.2 (a) and (b)).

$$\texttt{Area}_{int} = \int_{x_B}^{x_A} (\frac{b}{a}\sqrt{a^2 - (x - x_e)^2} + y_e + \sqrt{r_c^2 - (x - x_c)^2} - y_c)dx \tag{A.8}$$

• There are four intersection points between the virtual landmark $\texttt{P}$ and the bead $\texttt{B}_i$ (see Fig. A.2 (c)).

$$\begin{aligned}\texttt{Area}_{int} =& \pi r_c^2 - \int_{x_B}^{x_A} (\sqrt{r_c^2 - (x - x_c^2)} + y_c - \frac{b}{a}\sqrt{a^2 - (x - x_e)^2} - y_e)dx \\ & - \int_{x_B^{'}}^{x_A^{'}} (y_e - \frac{b}{a}\sqrt{a^2 - (x - x_e)^2} + \sqrt{r_c^2 - (x - x_c)^2} - y_c)dx\end{aligned} \tag{A.9}$$

In Fig. A.2, we can easily know that S is the earliest point that one MO can arrive at $\texttt{P}$, and E is the latest point that MO can leave $\texttt{P}$. Therefore, given the maximum speed v_{max}, we can calculate $\bar{t}^s$ and $\bar{t}^e$ as follows.

$$\bar{t}^s = t_i + \frac{|o_i S|}{v_{max}}, \quad \bar{t}^e = t_{i+1} - \frac{|o_{i+1} E|}{v_{max}} \tag{A.10}$$

Based on $\bar{t}^s$ and $\bar{t}^e$, we can calculate $\tau(\texttt{P}, \texttt{B}_i)$ by Eq. (A.5). After calculating $\texttt{R}_{int}$ and $\tau(\texttt{P}, \texttt{B}_i)$, given an uncertain trajectory necklace database D and a trajectory query $\texttt{T}_q$, we can compute $Sim_{cont}^{\texttt{P}\cap\texttt{B}_i}$ and $Sim_{cont}^{\texttt{P}\cap\texttt{T}}$ by Eq. (A.6) and (A.7), respectively.

Vita

Candidate's full name: Xichen Zhang
University attended:

- University of New Brunswick, Fredericton, Canada, 2019 - 2022, Ph.D. of Computer Science
- University of New Brunswick, Fredericton, Canada, 2015 - 2018, Master of Computer Science
- China National Pulp and Paper Research Institute, Beijing, China, 2010 - 2013, Master of Engineering
- Changsha University of Science and Technology, Changsha, China, 2006 - 2010, Bachelor of Engineering

Publications:

- **X. Zhang**, R. Lu, S. Zhang, S. Ray, A. Ghorbani, "Privacy-preserving Worker Selection in Mobile Crowdsensing over Spatial-temporal Constraints", in IEEE International Conference on Communications (ICC), pp. 1-6, IEEE, 2022.
- **X. Zhang**, R. Lu, S. Ray, J. Shao, A. Ghorbani, "Spatio-temporal Similarity based Privacy-preserving Worker Selection in Mobile Crowdsensing", in 2021 IEEE Global Communications Conferences (Globecom), pp. 1-6, IEEE, 2021.
- **X. Zhang**, R. Lu, J. Shao, F. Wang, H. Zhu, and A. Ghorbani, "FedSky: An Efficient and Privacy-preserving Scheme for Federated Mobile Crowdsensing." IEEE Internet of Things Journal. 9(7): 5344-5356 (2022).
- **X. Zhang**, S. Ray, F. Shoeleh, R. Lu, "Efficient Contact Similarity Query over Uncertain Trajectories", In the Proceedings of the 24th International Conference on Extending Database Technology (EDBT), pp. 403-408, 2021.
- **X. Zhang**, R. Lu, J. Shao, H. Zhu, and A. Ghorbani. "Continuous Probabilistic Skyline Query for Secure Worker Selection in Mobile Crowdsensing." IEEE Internet of Things Journal. 8(14): 11758-11772 (2021).

- **X. Zhang**, R. Lu, J. Shao, H. Zhu, and A. Ghorbani. "Secure and Efficient Probabilistic Skyline Computation for Worker Selection in MCS." IEEE Internet of Things Journal. 7(12): 11524-11535 (2020).
- **X. Zhang**, R. Lu, J. Shao, H. Zhu, and A. Ghorbani. "Achieve Secure and Efficient Skyline Computation for Worker Selection in Mobile Crowdsensing." In 2019 11th International Conference on Wireless Communications and Signal Processing (WCSP), pp. 1-6. IEEE, 2019.