SECURE AND PRIVACY-PRESERVING DATABASE-DRIVEN DYNAMIC SPECTRUM SHARING

by

Yidan Hu

A dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Spring 2021

SECURE AND PRIVACY-PRESERVING DATABASE-DRIVEN DYNAMIC SPECTRUM SHARING

by

Yidan Hu

Approved: _____
Kathleen F. McCoy, Ph.D.
Chair of the Department of Computer and Information Sciences

Approved: _____
Levi T. Thompson, Ph.D.
Dean of the College of Engineering

Approved: _____
Louis F. Rossi, Ph.D.
Vice Provost for Graduate and Professional Education and
Dean of the Graduate College

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Rui Zhang, Ph.D.
Professor in charge of dissertation

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Chien-Chung Shen, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Lena Mashayekhy, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Xiugang Wu, Ph.D.
Member of dissertation committee

I certify that I have read this dissertation and that in my opinion it meets the academic and professional standard required by the University as a dissertation for the degree of Doctor of Philosophy.

Signed: _____

Dejun Yang, Ph.D.
Member of dissertation committee

ACKNOWLEDGEMENTS

First of all, I would like to express my sincere gratitude to my advisor Dr. Rui Zhang for his great support and inspiring guidance during my Ph.D. study at the University of Delaware. Through his rigorous training, I have learned how to read research papers, identify and select challenging research problems, conceive novel ideas, turn my ideas into full technical solutions, setup experiments and simulations, and write well-organized research papers. His high level of expertise, warm encouragement, and professional guidance motivate me to pursue an academic career. Without him, I cannot successfully find a tenure-track faculty position in a research university, especially when most departments are cutting back in faculty hiring with the pandemic. I feel so lucky to be his student, and words are powerless to express my gratitude to him.

Second, I would like to thank my other committee members: Dr. Chien-Chung Shen, Dr. Lena Mashayekhy, Dr. Xiugang Wu, and Dr. Dejun Yang from Colorado School of Mines. Their expertise provide valuable guidance and comments to enrich my project. I also want to thank my labmates in the Network and Information Security Lab at the University of Delaware. They provide me with a friendly and constructive environment to carry out my Ph.D. research and help me with my research work. I am also very thankful to many faculty members and staff in our department, including, but not limited to, Dr. Kathleen McCoy, Dr. Hagit Shatkay, Dr. Lori Pollock, Dr. Sandra Carberry, Dr. Errol Lloyd, Ms. Teresa Twohig, and Ms. Kristi Halberg, for their help and support during my Ph.D. study. In addition, I would like to extend my thank to Dr. Phillips who shared their collected spectrum measurements with me for evaluating my proposed solutions.

My sincerely thank also goes to my lovely family. Without their encouragement and support, I cannot go to the United States and complete my Ph.D. degree.

Last but not least, this dissertation is based upon work supported in part by the National Science Foundation under grant No. CNS-1651954, CNS-1718078, and CNS-1933047. Many thanks for their generous support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Database-driven Dynamic Spectrum Sharing (DSS) is the de facto technical paradigm adopted by Federal Communications Commission (FCC) for meeting the ever-growing spectrum demand by allowing secondary users (SUs) to opportunistically access licensed spectrum bands without causing interference to primary users transmissions. In a database-driven DSS system, a geo-location database administrator (DBA) maintains the spectrum availability in its service region in the form of a radio environment map (REM). Maintaining accurate spectrum availability information requires the DBA to periodically collect a large number of spectrum measurements, for which a promising approach is to rely on mobile crowdsourcing by outsourcing spectrum sensing tasks to distributed mobile users. Database-driven DSS armed with crowdsourcing-based spectrum sensing, unfortunately, faces many security and privacy challenges.

This dissertation tackles three key security and privacy challenges in database-driven DSS to pave the way for its wide development and deployment. First, the DBA relies on spectrum measurements submitted by mobile users to construct and maintain the REM, but some mobile users may be malicious or compromised to submit false spectrum measurements. To tackle this challenge, we introduce a novel mechanism for secure REM construction in the presence of false measurements. Second, crowdsourcing-based spectrum sensing relies on mobile users' participation, who not only require strong incentive, but also demand privacy protection. To tackle this challenge, we design an incentive mechanism that simultaneously achieves differential bid privacy, truthfulness, and high REM accuracy. Third, an effective approach to process a large number of spectrum access requests with low latency is to adopt the edge computing paradigm by having the DBA continuously pushes the spectrum availability

updates to distributed local edge servers, which in turn process spectrum access requests from nearby SUs on the DBA's behalf. However, edge servers owned by different entities cannot be fully trusted to process SU's spectrum request based on authentic and the most recent spectrum information, which may result in either loss of revenue or harmful interference to PUs' transmissions. To tackle this challenge, we propose a novel freshness authentication mechanism to allow SUs to verify that their spectrum-access requests are decided based on authentic and up-to-date spectrum availability information.

Chapter 1

INTRODUCTION

The deep penetration of smartphones and tablets into people's everyday life along with the explosive growth in mobile apps have created an ever-increasing demand for wireless spectrum. On the one hand, most of the usable radio spectrum has already been licensed to various government and commercial entities. On the other hand, several studies [1, 2, 3] have shown that many licensed spectrum allocated to government and military entities are highly underutilized. The need for enhancing wireless spectrum access has been highlighted as a catalyst for economic growth in FCC's National Broadband Plan [2] and the President's Council of Advisors on Science and Technology (PCAST) report [3], which call for fundamental paradigm shifts and novel technologies to take full advantage of the underutilized licensed spectrum.

Database-driven Dynamic Spectrum Sharing(DSS) [4, 5] is the de facto technical paradigm adopted by Federal Communications Commission (FCC) for improving spectrum utilization by allowing secondary users (SUs) with cognitive radio(CR) capabilities to opportunistically access licensed spectrum bands without interrupting the transmissions of licensed primary users (PUs). In such a system, a geo-location database administrator (DBA) maintains the spectrum availability in its service region. Any SU is required to inquire the DBA about the availability of any interested spectrum before using it. The DBA either grants or denies the SU's spectrum-access request based on the maintained spectrum availability at the desired time and location.

In the current proposal, the DBA estimates the spectrum availability based on the registered locations and transmission schedules of primary users (PUs) in combination with radio propagation modeling, e.g., FCC Curves [6] based on the Longley-Rice model [7]. Recent measurement studies [8, 9, 10, 11], however, have shown that such

estimations are often inaccurate and tend to be overly conservative for ignoring local environmental factors (e.g., trees and high-rise buildings), resulting in a considerable waste of valuable spectrum opportunities. A more promising approach to enhance the spectrum availability estimation accuracy is to let the DBA construct and maintain a Radio Environmental Map (REM) [12], where the PU's received signal strength (RSS) at every location of interest is either directly measured or estimated using proper statistical spatial interpolation techniques.

Constructing and maintaining an accurate REM requires periodically collecting a large number of spectrum measurements over the DBA's service region. A widely advocated approach is to let the DBA deploy a small number of dedicated spectrum sensors at strategic locations [8, 9] and outsource the majority of spectrum-sensing tasks to ubiquitous mobile users [13, 14]. The feasibility of this outsourcing approach lies in two main aspects. First, mobile devices penetrate deeply into people's everyday life which indicates sufficient geographic coverage especially in metropolitan areas where the spectrum demand is the highest. Second, future mobile devices are widely expected to be capable of spectrum sensing via either internal spectrum sensors or external ones acquired from the DBA [8, 9]. Crowdsourcing-based spectrum sensing is much more cost effective than deploying a large-scale dedicated spectrum sensor network, which is well known to be prohibitive to deploy and difficult to maintain.

Security and privacy concerns are among the most challenging obstacles to the wide deployment of database-driven DSS systems armed with spectrum-sensing outsourcing. For example, mobile users may be malicious or compromised to submit false spectrum measurements. Despite significant efforts on coping with such false-data injection attacks in DSS systems [15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26], how to construct a sufficiently accurate REM in the presence of forged spectrum measurements remains an open challenge. As another example, mobile users not only require strong incentives for participating in spectrum sensing, but also demand adequate privacy protection. How to design a sound incentive mechanism for crowdsourcing-based spectrum sensing is unclear. In view of these challenges, this dissertation aims to tackle

the following three key security and privacy challenges in database-driven DSS to pave the way for its wide development and deployment.

- **Secure crowdsourced REM construction**. Crowdsourcing-based spectrum sensing is a promising approach for the DBA to construct and maintain an REM in its service region. However, mobile users may be malicious or compromised to submit false spectrum measurements. Since REM construction commonly relies on statistical interpolation techniques [27, 28, 29, 30, 13] that are known to be sensitive to outliers, even a small number of false measurements would significantly degrade the accuracy of the REM, leading to either missed spectrum opportunities or harmful interference to PU's transmissions. To tackle this challenge, we have developed ST-REM, a novel framework for secure crowdsourced REM construction in the presence of false spectrum measurements. Inspired by the self-labeled techniques [31], ST-REM constructs highly accurate REMs from a small number of trusted measurements and many more untrusted measurements via iterative statistical spatial interpolation.
- **Incentive-compatible and differentially-private spectrum sensing**. Strong incentive is needed to stimulate mobile users to participate in crowdsourcing-based spectrum sensing. However, self-interested mobile users may game the system to earn extra credits they are not entitled to. In addition, mobile users may also hesitate to participate if their sensitive information is not adequately protected. To tackle this challenge, we propose a novel incentive mechanism, DPS, that simultaneously achieves differential bid privacy, approximated truthfulness, and high REM accuracy.
- **Secure edge computing-based spectrum access request processing**. A promising approach for reducing spectrum-access request processing latency is to explore the emerging edge computing paradigm by having the DBA proactively push spectrum availability updates to distributed local edge servers, which in turn process spectrum-access requests from nearby SUs on the DBA's behalf. However, edge servers, commonly owned by different entities, cannot be fully trusted to process spectrum-access requests based on the most recent spectrum

availability, which may lead to either loss of revenue for spectrum owners or harmful interference to PUs' transmissions. To tackle this challenge, we propose to design a novel freshness authentication technique to allow any SU to verify whether the decision made by the local edge server is based on the most recent spectrum availability information from the DBA.

1.1 Organization

The remainder of this dissertation is structured as follows. In Chapter 2, we introduce ST-REM, a novel scheme for securely constructing REMs in the presence of false spectrum measurements. In Chapter 3, we present a differentially-private incentive mechanism for stimulating mobile users' participation in crowdsourced spectrum sensing while protecting their bid privacy. In Chapter 4, we propose KV-Fresh, a novel freshness authentication mechanism allowing SUs to verify whether their spectrum-access requests are processed based on authentic and most up-to-date spectrum availability information from the DBA. We finally conclude this dissertation in Chapter 5.

Chapter 2

SECURE CROWDSOURCED RADIO ENVIRONMENT MAP CONSTRUCTION

2.1 Introduction

Database-driven Dynamic Spectrum Sharing (DSS) is the de facto technical paradigm adopted by Federal Communications Commission (FCC) for meeting the ever-growing spectrum demand by allowing SUs to opportunistically access licensed spectrum bands without causing interference to PUs' transmissions [4, 5]. In a database-driven DSS system, a geo-location database administrator (DBA) maintains the spectrum availability in its service region and manages spectrum access from secondary users. Any SU who wants to access a licensed spectrum band is required to inquire the DBA, which may either grant or deny the spectrum-access request based on the spectrum availability at the desired time and location.

Effectively enhancing spectrum utilization requires accurate spectrum availability information, for which a widely advocated approach is to let the DBA construct and maintain a Radio Environmental Map (REM) over its service region. The REM concept [32, 33] was originally proposed as an abstraction of radio environments represented by a distributed database for storing information and knowledge of the radio environment to support a wide range of spectrum-related functionalities. Following the recent work [34], we consider an REM as a map characterizing primary users' radio activities, in which the primary users' received signal strength (RSS) in every location of interest is either directly measured via spectrum sensing or estimated using proper statistical spatial interpolation techniques.

Maintaining an accurate REM requires the DBA to periodically collect many spectrum measurements over a large geographic region, which can be accomplished

in mainly two ways. A straightforward approach is to deploy a network of spectrum sensors for detecting radio activities on licensed spectrum bands. However, it is well known that large-scale sensor networks are expensive to deploy and difficult to operate and maintain. Therefore, it has been widely advocated that the DBA only needs to deploy a small number of dedicated spectrum sensors at strategic locations [8, 9] and outsource the majority of spectrum-sensing tasks to ubiquitous mobile users. The feasibility of this approach lies in the deep penetration of mobile devices into everyday life and the wide expectation that future mobile devices can perform spectrum sensing via either internal spectrum sensors or external ones acquired from other parties like the DBA [18, 21, 19, 35, 36, 37, 38].

Crowdsourcing-based REM construction is, unfortunately, vulnerable to false spectrum measurements, which contain RSS values much higher (or lower) than the true RSS values. In particular, mobile users cannot be fully trusted and may submit false spectrum measurements due to various reasons. For example, a good mobile user may submit false spectrum measurements because of faulty spectrum sensor. As another example, a selfish mobile user may submit forged spectrum measurements to claim the reward at the DBA without actual sensing to save battery. Last but not least, a malicious mobile user may be hired by the DBA's business competitor to submit false spectrum measurements to damage the DBA's reputation. Since most existing approaches for REM construction [27, 28, 29, 30, 13] rely on statistical interpolation techniques, e.g., Ordinary Kriging, that are known to be sensitive to outliers [39], even a small number of false measurements can heavily distort the REM, leading to either missed spectrum opportunities or harmful interference to PUs' transmissions.

Despite the large body of work on secure cooperative spectrum sensing against false spectrum measurements [15, 16, 17, 18, 19, 20, 21, 22, 23], how to construct an accurate REM from possibly false spectrum measurements poses new challenges. In particular, secure cooperative sensing aims to decide whether a primary user at a known location is transmitting or not, whereas REM construction intends to estimate the primary user's RSS at every location of interest where the primary user's transmission

activity is known. The unique challenges brought by REM construction render prior solutions [15, 16, 17, 18, 19, 20, 21, 22, 23] inapplicable. These situations call for sound solutions to construct REM with high accuracy in the presence of false spectrum measurements.

To tackle this challenge, we introduce ST–REM, a novel spatiotemporal approach for securely constructing REMs in the presence of false spectrum measurements. Inspired by self-labeled techniques [31] originally developed for semi-supervised learning, our proposed approach constructs highly accurate REMs from a small number of trusted measurements and many more untrusted measurements via iterative statistical spatial interpolation. Specifically, an initial REM is constructed using only the trusted measurements from dedicated spectrum sensors and then gradually refined by incorporating the most trustworthy measurements from the remaining ones. The key ingredient of the proposed approach is a novel mechanism for evaluating of the trustworthiness of every spectrum measurements submitted by mobile users, which jointly considers the measurement's spatial and temporal trustworthiness. The former is evaluated based on the measurement's spatial fitness with other measurements that have already been deemed trustworthy. The latter, on the other hand, is evaluated by tracking the mobile user's long-term behavior, which provides strong indication for the quality of the measurement he/she submits in the current epoch. Using the most trustworthy spectrum measurements, the DBA is able to filter out false ones and construct an REM with high accuracy. Our contributions in this chapter can be summarized as follows.

- To the best of our knowledge, we are the first to study secure crowdsourced REM construction in the presence of false spectrum measurements.
- We introduce ST–REM, a novel approach for constructing REM from a small number of trusted measurements from dedicated spectrum sensors and many more from untrusted mobile users. The accuracy of the resulting REM is achieved

by jointly considering the spatial and temporal trustworthiness of the measurements from mobile users and constructing the REM only using the most trustworthy ones.

- The efficacy of ST–REM is confirmed via extensive simulation studies using a real spectrum measurement dataset. For example, our simulation results show that even when twenty percent of the measurements are false, ST–REM can produce an REM with mean absolute error (MAE) of 2.75dB, which is only 2.83% higher than the case where all false measurements are known in advance and excluded by the DBA.

The rest of this chapter is structured as follows. Related work is discussed in Section 2.2. We introduce the system and adversary models along with the design goals in Section 2.3. Section 2.4 presents the design of ST–REM. We evaluate the performance of the proposed approach in Section 2.5. Section 2.6 concludes our work.

2.2 Related Work

In this section, we discuss prior work in several areas that are most germane to our work.

2.2.1 REM Construction via Statistical Spatial Interpolation

There have been a number of attempts to improve the spectrum estimation accuracy at the DBA by constructing an REM or detailed PU coverage map from spectrum measurements through statistical spatial interpolation, for which a recent survey can be found at [40].

Ordinary Kriging is the most popular statistical spatial interpolation technique for radio mapping. Alaya-Feki *et al.* [27] introduced a solution for constructing a map of received signal strength from radio measurements using Ordinary Kriging. In [28], Achtzehn *et al.* conducted a large-scale measurement campaign and demonstrated that spatial interpolation techniques such as Ordinary Kriging outperform well-known propagation models in predicting transmitter's signal strengths in the TV whitespace.

Another measurement study was reported in [29], in which Phillips *et al.* used Ordinary Kriging to estimate the coverage of a 2.5 GHz WiMax network in a US university campus. A similar study appeared in [41], which showed that the accuracy of TVWS geo-location database can be improved by predicting the primary user's signal strength with a relatively small number of measurements using Ordinary Kriging. The advantage of Ordinary Kriging over model-based prediction such as Longley-Rice model, FCC F-Curves, and k nearest neighbor, is later reconfirmed by another measurement study in Seattle, WA in [30]. Crowdsourcing-based REM construction using Ordinary Kriging was firstly studied in [13], in which Ying *et al.* introduced an incentive mechanism to stimulate mobile users' participation.

Other statistical spatial interpolation techniques have also been used for radio mapping. Ojaniemi *et al.* explored several methods, including Ordinary Kriging, Cokriging, and spatial simulated annealing, for integrating field measurements into radio propagation model [42]. Dai *et al.* [43] proposed a framework for integrating spectrum sensing results and spectrum database via Delaunay triangulation. Delaunay triangulation was also used in [41] to predict the signal strengths at unmeasured locations.

All these works assume that all the measurements are trusted, while it is well known that these statistical spatial interpolation techniques are sensitive to outliers due to masking and swamping effects. For example, it was shown in [39] that even a small number of false measurements could significantly affect the predictions at unobserved locations.

2.2.2 Secure Cooperative Spectrum Sensing

Secure cooperative spectrum sensing has been studied extensively in the past few decades, where the goal is to determine whether or not a PU at a known location is transmitting from potentially false spectrum measurements. Existing solutions can be generally classified into three categories.

The first category detects and filters out false spectrum measurements via statistical anomaly detection. In [16], Min *et al.* proposed an attack-tolerant distributed

sensing protocol by exploring shadow fading correlation to detecting abnormal spectrum sensing results. A Bayesian-based approach was introduced in [44] to evaluate the suspicious level of spectrum sensing reports whereby to filter out potential false ones. In [25], Wang *et al.* introduced a joint spectrum sensing and access framework based on statistical hypothesis testing to cope with false spectrum sensing reports. A secure cooperative spectrum sensing scheme was introduced in [45] to detect false sensing reports with M-ary quantized sensing data.

The second category uses reputation system to track sensors' long term behaviors to differentiate bad sensors from good ones. Typically, every sensor's reputation score is computed based on the accuracy of their past sensing measurements [15] or whether its local decision matches the global network decision [20]. A sensor is considered misbehaving if its reputation score drops below a certain threshold. For example, a reputation-based detection scheme is introduced in [46] in which sensing reports from a sensor would be excluded from the fusion process if its reputation score exceeds certain threshold. More recently, reputation score is incorporated into learning process to determine possible punishment for secondary users with poor sensing performance [47].

The third category relies on machine learning techniques to differentiate false measurements from good ones. In [19], the authors proposed to train a classifier using Support Vector Machines from reliable sensing reports whereby to detect and filter false spectrum measurements. A reinforcement-learning-based user selection method is proposed in [47] to select secondary according to their past performance.

Finally, it has been shown in [22, 23] that trusted sensors can be used to defend against false measurements. For example, PUET [22] is a technique that explores a trusted transmitter transmitting test signals to detect sensing data falsification attacks. Reputation-based mechanisms have also been integrated with trusted users in [20, 23]. Furthermore, trusted measurements are also used as training data for machine learning solutions [19].

As discussed in Section 2.1, none of these solutions can be applied to the problem

of secure REM construction, in which the PU's location and transmission activity are known but its signal strength needs to be estimated at every location of interest.

2.2.3 False Data Injection Attack in Crowdsensing System

False data injection attack has also been studied in general crowdsensing systems. For example, Yang *et al.* [48] introduced an unsupervised learning approach to evaluate users' sensing qualities and long-term reputations and filter out anomalous sensing data. A scheme was proposed in [49] to enhance data trustworthiness in crowdsourcing-based positioning systems. More recently, data poisoning attacks were studied in [50, 51], where Miao *et al.* introduced several attack strategies to allow malicious workers to maximize attack utility while evading detection. None of these works can be directly applied for secure crowdsourced REM construction.

2.3 Preliminary

In this section, we first introduce the system and adversary models and then our design goals.

2.3.1 System Model

Fig. 2.1 shows an exemplary database-driven DSS system. We consider a DBA that provides spectrum access service to secondary users in its service region $\mathcal{D}$.

The DBA estimates the spectrum availability through spectrum sensing and constructing and periodically updating an REM over $\mathcal{D}$. As in [21, 23], we assume that the DBA deploys a small number of stationary spectrum sensors at strategic locations, referred to as *anchor sensor*s hereafter. Anchor sensors can be remotely attested by the DBA and excluded if they are detected as compromised. Due to cost constraints, the DBA cannot afford to deploy too many anchor sensors to cover the entire service region and still relies on the spectrum measurements from the majority of mobile users, referred to as *mobile sensor*s hereafter to ensure the accuracy of the REM. We subsequently denote by Θ_a the set of anchor sensors and Θ_m the set of mobile sensors.

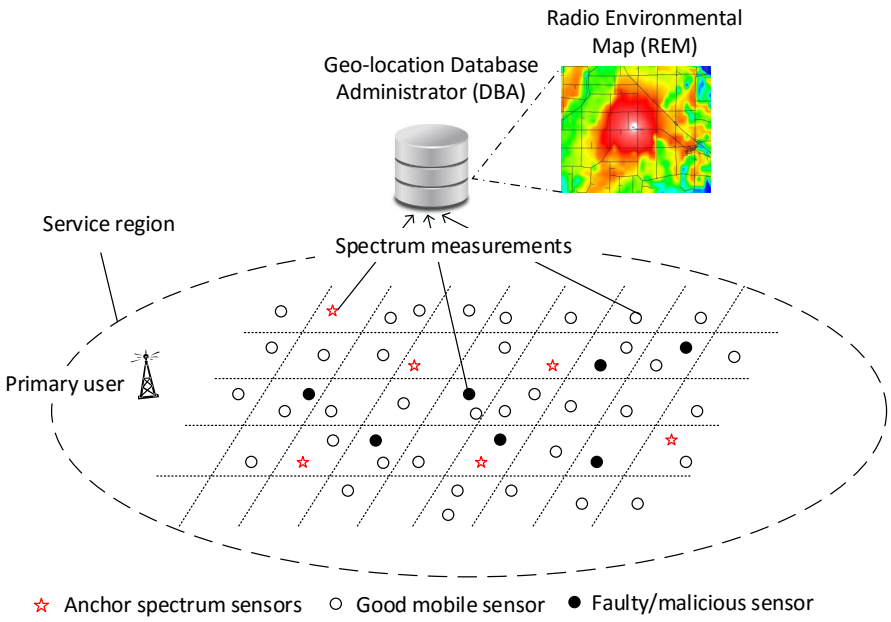


Figure 2.1: An exemplary database-driven DSS system.

We assume that the time is divided into epochs of equal length. At the beginning of each epoch, every sensor $i \in \Theta_a \bigcup \Theta_m$ submits a spectrum measurement $R_i = (Z_i, \mathbf{x}_i)$, where Z_i is the measured RSS (in dBm) at location $\mathbf{x}_i$. We assume that the service region $\mathcal{D}$ is divided into N non-overlapping *cell*s of equal size. Some cells may not have any measurements taken, and the locations at which measurements are taken may not be the center of any cell. Given the set of spectrum measurements $\mathcal{R} = \{R_i | i \in \Theta_a \bigcup \Theta_m\}$, the DBA's goal is to construct an REM by estimating the RSS at the center of every cell.

To ease the presentation, we assume that there is one primary user in $\mathcal{D}$ whose location and transmission schedule are known to the DBA. Our work, however, can be easily adapted to support multiple primary users with minimal effort.

2.3.2 Adversary Model

The DBA is trusted to faithfully perform all system operations, and the spectrum measurements submitted by anchor sensors are trusted. In contrast, mobile sensors may submit false spectrum measurements due to faculty spectrum sensors, forging spectrum measurements to claim the reward at the DBA without actual sensing, or being hired by the DBA's business competitor to damage its reputation. We assume that false spectrum measurements may contain RSS values arbitrarily different from the true RSS measurements and that the number of false measurements is unknown to the DBA in advance. On the other hand, we do not specifically consider spectrum measurements with forged locations because such measurements are equivalent to false measurements at the claimed locations. We assume that the attacker can submit false RSS measurements in different epochs following an arbitrary strategy unknown to the DBA.

Our subsequent discussion focuses on REM construction in the presence of false spectrum measurements. We assume that communications between anchor/mobile sensors and the DBA are properly secured via standard cryptographic techniques such as TLS [52]. Moreover, we do not consider other attacks targeting DSS systems such as primary user emulation attack for which we resort to existing rich literatures such as [53].

2.3.3 Designed Goals

The proposed approach is designed with the following goals in mind.

- *Resilience against false spectrum measurements*: The approach should produce an REM in the presence of unknown number of false spectrum measurements with high accuracy. In particular, it should produce an REM with much higher accuracy than either using only trusted spectrum measurements from anchor sensors or blindly using all spectrum measurements.
- *Low deployment cost*: The proposed approach should only require a small number of anchor sensors to ensure sufficiently high accuracy of the resulting REM.

2.4 ST–REM: A Spatiotemporal Approach

In this section, we first give an overview of the proposed spatiotemporal approach and introduce the background of Ordinary Kriging, the interpolation technique used by the proposed approach. We then detail the design of our proposed approach.

2.4.1 Overview

ST–REM is designed to construct highly accurate REMs using a small number of trusted measurements and many untrusted measurements via iterative statistical spatial interpolation. This approach is inspired by the self-labeled techniques [31] proposed for semi-supervised learning with the goal of exploring a small amount of labeled data and a large amount of unlabeled data for classification [31]. In self-labeled techniques, an initial classifier is trained based on the labeled data only, which is then applied to the unlabeled data to generate more labeled samples as additional input to refine the classifier. Self-labeled techniques have been shown to surpass the classification performance achieved by either supervised learning where all unlabeled data are discarded or unsupervised learning where all label information is ignored.

As an analogue to self-labeled techniques, the proposed approach constructs an REM in an iterative fashion. In each epoch, on receiving all the spectrum measurements, an initial REM is constructed using only the trusted measurements from anchor sensors. In each subsequent iteration, a fixed number of remaining measurements deemed most trustworthy are incorporated to refine the REM. This process continues until certain terminal condition is met, at which point all remaining measurements are discarded and the final REM is produced.

A key component of the proposed approach is the evaluation of the trustworthiness of measurements submitted by mobile users. In particular, our proposed approach calculates a spatial trustworthiness score and a temporal trustworthiness score for every measurement. The spatial trustworthiness score is computed based on the measurement's spatial fitness with the REM constructed from the measurements that have already been deemed trustworthy. The temporal trustworthiness score, on the other

hand, is computed from the mobile sensor's past performance, which provides strong indication for the quality of the measurement he/she submits in the current epoch. The overall trustworthiness score of the measurement is obtained by combining its spatial and temporal trustworthiness scores.

While the proposed approach is general in the sense that it can be integrated with different statistical interpolation techniques, we take Ordinary Kriging [54] as an example to illustrate its design for Ordinary Kriging's overwhelming popularity and satisfactory performance in REM construction [27, 28, 29, 13, 55, 56, 30]. In what follows, we first briefly introduce the background of Ordinary Kriging and then detail the design of the proposed spatiotemporal approach.

2.4.2 Background on Ordinary Kriging

Kriging [54] is a class of geo-statistical spatial interpolation techniques originally developed for mining but have been increasingly being used for radio mapping. Under Kriging, the RSS at any location $\mathbf{x}$ is modeled as a Gaussian random field in the form

$$Z(\mathbf{x}) = \mu(\mathbf{x}) + \delta(\mathbf{x}),$$

where $\mu(\mathbf{x})$ is the mean capturing path loss and shadowing, and $\delta(\mathbf{x})$ represents possible sampling error.

In Ordinary Kriging [54], $Z(\mathbf{x})$ is further assumed to be *intrinsic stationary* in the sense that

$$\begin{aligned} &\mathsf{E}[Z(\mathbf{x})] = \mu(\mathbf{x}) = \mu, \\ &\mathsf{E}[(Z(\mathbf{x}_1) - Z(\mathbf{x}_2))^2] = 2\gamma(h) \ , \end{aligned} \tag{2.1}$$

for all $\mathbf{x} \in \mathcal{D}$, where $\mathsf{E}[\cdot]$ denotes expectation, μ is an unknown constant, $h = ||\mathbf{x}_1 - \mathbf{x}_2||$ is the *distance lag* between two locations $\mathbf{x}_1$ and $\mathbf{x}_2$, and $\gamma(\cdot)$ is the *semivariogram* function that models the variance between two locations as a function of their distance. The assumption of intrinsic stationarity may not hold for spectrum measurements but has been found acceptable in the literature [27, 28, 13, 55, 56, 30], especially

after removing possible source of nonlinear trend from measurements through a proper detrending process [29].

2.4.3 Detailed Design of ST–REM

In each epoch, the DBA constructs an REM from the set of measurements $\mathcal{R} = \{R_i | i \in \Theta_a \bigcup \Theta_m\}$ it receives in three steps. First, the DBA performs detrending process to the measurements to remove possible nonlinear trend from the measurements so that the residue measurements fit the Ordinary Kriging model better. Second, the DBA constructs an REM from the detrended measurements in an iterative fashion. Finally, the DBA adds the detrended values back to produced REM to generate a final REM.

2.4.3.1 Detrending

Detrending is the process of removing any non-linear trend from the measurement original spectrum measurements, which is usually preferred as the resulting detrended measurements would better fit the Ordinary Kriging model [57]. Specifically, given an original spectrum measurement $R_i = (Z_i, \mathbf{x}_i)$ from a mobile sensor or an anchor sensor, the corresponding detrended measurement is given by $R'_i = (S_i, \mathbf{x}_i)$, where

$$S_i = Z_i - P(\mathbf{x}_i)$$

is the residue RSS at $\mathbf{x}_i$ and $P(\mathbf{x}_i)$ is the RSS at $\mathbf{x}_i$ predicted by a suitable model. ST–REM does not rely on any specific detrending procedure but assumes the existence of a suitable one for the received measurements. We will present an exemplary detrending procedure adopted from [29].

2.4.3.2 Iterative Measurement Selection

Given the set of detrended measurements $\{R'_i | i \in \Theta_t \bigcup \Theta_c\}$, the DBA gradually selects a set of measurements in an iterative fashion for REM construction. Specifically, the DBA maintains a trusted sensor set Θ_t and a candidate sensor set Θ_c at all time,

where $\Theta_t = \Theta_a$ and $\Theta_c = \Theta_m$ initially. In each iteration, the DBA does the following in sequel.

First, for every candidate measurement $R'_j, j \in \Theta_c$, the DBA calculates a trust score T_j. The process of calculating T_j is deferred to Section 2.4.3.3. Second, the DBA finds the q measurements with the highest trust scores, denoted by Θ_q, where q is a system parameter that represents the tradeoff between computation overhead and accuracy of the final REM. Third, the DBA moves Θ_q to the trusted sensor set, i.e., $\Theta_t = \Theta_t \bigcup \Theta_q$ and $\Theta_c = \Theta_c \setminus \Theta_q$.

The selection process is terminated if the ratio between the number of trusted measurements and the total number of measurements reaches a predetermined threshold η, i.e.,

$$\frac{|\Theta_t|}{|\Theta_a \bigcup \Theta_m|} \geq \eta \ ,$$

where η is a system parameter. All the remaining candidate measurements $\{R'_j | j \in \Theta_c\}$ are then discarded.

2.4.3.3 Spatiotemporal Trustworthiness Evaluation

A key component of ST–REM is a novel method to evaluate the trustworthiness of a candidate measurement by jointly considering its spatial fitness with other trusted measurements and the sensor's past performance. Specifically, for every candidate measurement $R'_j, j \in \Theta_c$, the DBA calculates a spatial trust score and a temporal trust score and then combines the two into an overall trust score.

Spatial trust score. The spatial trust score of a measurement $R'_j, j \in \Theta_c$, characterizes its spatial fitness with current trusted measurements $\{R'_i | i \in \Theta_t\}$. The key idea is to construct an REM using the current trusted measurements whereby to predict the RSS value at the candidate measurement's location $\mathbf{x}_j$. The smaller the difference between the reported RSS value and the predicted RSS value, the better R'_j fits the current trusted measurements, the more trustworthy of the candidate measurement, and vice versa. In particular, the spatial trust score of each measurement R'_j is calculated as follows.

First, the DBA builds an empirical semivariogram $\hat{\gamma}(h)$ from the current trusted measurement set $\{R'_i | i \in \Theta_t\}$. Specifically, the DBA computes

$$\hat{\gamma}(h) = \frac{1}{2|\mathcal{P}(h)|} \sum_{(\mathbf{x}_i, \mathbf{x}_k) \in \mathcal{P}(h)} (S_i - S_k)^2,$$

where $\mathcal{P}(h) = \{(\mathbf{x}_i, \mathbf{x}_k) | i, k \in \Theta_t, ||\mathbf{x}_i - \mathbf{x}_k|| = h\}$ is the set of location pairs with distance h. The DBA then fits $\hat{\gamma}(h)$ with a suitable parametric model. There are several popular parametric models in Ordinary Kriging, such as Gaussian, Cauchy, and Spherical models [58]. In this chapter, we choose the commonly used exponential model, which is given by

$$\gamma(h; \alpha_1, \alpha_2) = \alpha_1 (1 - \exp(\frac{-h}{\alpha_2})) \ ,$$

where α_1 is related to the variance of the spectrum measurements, and α_2 scales the correlation distance of the model. These parameters can be obtained from the estimated semivariogram via least squares estimator.

Second, the DBA estimates the residue RSS at location $\mathbf{x}_j$ at which candidate measurement R'_j was taken using the empirical semivariogram model $\hat{\gamma}(\cdot)$. Specifically, the DBA predicts the residue RSS at location $\mathbf{x}_j$ as a linear combination of the trusted residual measurements $\{R'_i | i \in \Theta_t\}$ given by

$$\hat{S}(\mathbf{x}_j) = \sum_{i \in \Theta_t} w_i \cdot S_i \ , \tag{2.2}$$

where $\sum_{i \in \Theta_t} w_i = 1$ are normalized weights. The estimation error is given by

$$\begin{aligned} \epsilon(\mathbf{x}_j) &= \hat{S}(\mathbf{x}_j) - S(\mathbf{x}_j) \\ &= (w_1, \ldots, w_{|\Theta_t|}, -1) \cdot (S_1, \ldots, S_{|\Theta_t|}, S(\mathbf{x}_j)) \ , \end{aligned}$$

where $S(\mathbf{x}_j)$ is the true RSS residue at $\mathbf{x}_j$ that may be different from the reported

residue S_j. It is easy to see that the above estimator is unbiased as

$$\begin{aligned}
\mathsf{E}[\epsilon(\mathbf{x}_j)] &= \sum_{i\in\Theta_t} w_i S(\mathbf{x}_i) - \mathsf{E}[S(\mathbf{x}_j)] \\
&= \sum_{i\in\Theta_t} w_i \mathsf{E}[S(\mathbf{x}_i)] - \mathsf{E}[S(\mathbf{x}_j)] \\
&= \sum_{i\in\Theta_t} w_i \mu - \mu \\
&= 0.
\end{aligned}$$

Let $h_{i,k} = ||\mathbf{x}_i - \mathbf{x}_k||$ for all $i, k \in \Theta_t$ and $h_{i,j} = ||\mathbf{x}_i - \mathbf{x}_j||$ for all $i \in \Theta_t, j \in \Theta_m$. Since minimizing the prediction variance of an unbiased predictor is equivalent to minimizing the mean squared error, we have

$$\begin{aligned}
\mathsf{Var}[\epsilon(\mathbf{x}_j)] &= \mathsf{E}[(\hat{S}(\mathbf{x}_j) - S(\mathbf{x}_j))^2] \\
&= \sum_{i\in\Theta_t}\sum_{k\in\Theta_t} w_i w_k \mathsf{E}[S(\mathbf{x}_i)S(\mathbf{x}_k)] - 2\sum_{i\in\Theta_t} w_i \mathsf{E}[S(\mathbf{x}_i)S(\mathbf{x}_j)] + \mathsf{E}[(S(\mathbf{x}_j))^2] \\
&= -\frac{1}{2}\sum_{i\in\Theta_t}\sum_{k\in\Theta_t} w_i w_k \mathsf{E}[(S(\mathbf{x}_i) - S(\mathbf{x}_k))^2] + \sum_{i\in\Theta_t} w_i \mathsf{E}[(S(\mathbf{x}_i) - S(\mathbf{x}_j))^2] \\
&= -\sum_{i\in\Theta_t}\sum_{k\in\Theta_t} w_i w_k \hat{\gamma}(h_{i,k}) + 2\sum_{i\in\Theta_t} w_i \hat{\gamma}(h_{i,j})
\end{aligned}$$

To find the optimal weights $\{w_i\}_{i\in\Theta_t}$, the DBA solves the following optimization problem

$$\begin{aligned}
\text{mininize} \quad & -\sum_{i\in\Theta_t}\sum_{k\in\Theta_t} w_i w_k \hat{\gamma}(h_{i,k}) + 2\sum_{i\in\Theta_t} w_i \hat{\gamma}(h_{i,j}), \\
\text{subject to} \quad & \sum_{i\in\Theta_t} w_i = 1.
\end{aligned}$$

The Lagrangian associated with the optimization problem is given by

$$\mathcal{L}(\mathbf{w}, \nu) = -\sum_{i\in\Theta_t}\sum_{k\in\Theta_t} w_i w_k \hat{\gamma}(h_{i,k}) + 2\sum_{i\in\Theta_t} w_i \hat{\gamma}(h_{i,j}) + \nu(\sum_{i\in\Theta_t} w_i - 1),$$

where ν is the Lagrange multiplier. Taking the partial derivatives of $\mathcal{L}(\mathbf{w}, \nu)$ with respect to the $\{w_i\}_{i\in\Theta_t}$ and ν, we can obtain

$$\begin{cases}
\frac{\partial \mathcal{L}}{\partial w_i} = 0, \quad \forall i \in \Theta_t, \\
\frac{\partial \mathcal{L}}{\partial \nu} = 0.
\end{cases}$$

The solution to the above optimization problem is then given by

$$\begin{pmatrix} w_1 \\ \vdots \\ w_{|\Theta_t|} \\ \nu \end{pmatrix} = \begin{pmatrix} \gamma(h_{1,1}) & \dots & \gamma(h_{1,|\Theta_t|}) & 1 \\ \vdots & \ddots & \vdots & \vdots \\ \gamma(h_{|\Theta_t|,1}) & \dots & \gamma(h_{|\Theta_t|,|\Theta_t|}) & 1 \\ 1 & \dots & 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} \gamma(h_{1,j}) \\ \vdots \\ \gamma(h_{|\Theta_t|,j}) \\ 1 \end{pmatrix} . \tag{2.3}$$

Under the optimized weights given in Eq. (2.3), the difference between the reported RSS value S_j and predicted RSS value S_j is given by $|\sum_{i\in\Theta_t} w_i S_i - S_j|$. Intuitively, the smaller the difference, the better measurement R'_j fits with other trusted measurements $\{R'_i | i \in \Theta_t\}$, and vice versa. Let $\epsilon_{\max}$ be the maximum estimation error, which we set to be the maximum detrended RSS among all anchor sensors, i.e., $\max\{S_j | j \in \Theta_a\}$. We define the *spatial trust score* of the measurement R_j (or corresponding detrended measurement R'_j) as

$$T_j^s = \frac{|\sum_{i\in\Theta_t} w_i S_i - S_j|}{\epsilon_{\max}} , \tag{2.4}$$

where $\{w_i\}_{i\in\Theta_t}$ is given in Eq. (2.3).

Temporal trust score. Unlike spatial trust score that considers a measurement's spatial fitness with other trusted measurements, the temporal trust score of a candidate measurement captures the mobile sensor's long-term behavior. As a mobile sensor participates in spectrum sensing in many epochs, its past performance can provide strong indication for the quality of spectrum measurement it submits in the current epoch. Recall that the DBA gradually incorporates candidate spectrum measurements into trusted measurement sets to construct the REM in each epoch. Intuitively, the earlier a measurement is added into the trusted measurement set, the better the measurement fits with existing trusted measurements, the higher quality of the measurement, and vice versa.

Based on the above intuition, the DBA maintains a *temporal trust score* T_j^t for each mobile sensor $j \in \Theta_m$, where $0 \le T_j^t \le 1$. When each mobile sensor j first joins the system, the DBA assigns an initial temporal score $T_j^t = \eta$, as the DBA does not know

whether or not its first measurement would be added to the trusted measurement set when iterative measurement selection terminates. At the end of each subsequent epoch, the DBA updates T_j^t based on the quality of measurement he submits. Consider epoch t as an example. Assume that measurement R_j from sensor j is the r_jth measurement moved from the candidate measurement set to the trusted measurement set, where we postulate that $r_j = |\Theta_m|$ if measurement R_j is discarded in the end. The DBA updates mobile sensor j's temporal trust score as

$$T_j^t = \alpha T_j^t + (1-\alpha)\frac{r_j}{|\Theta_m|} , \tag{2.5}$$

where $\alpha \in [0,1]$ is a system parameter that controls how fast past performance is forgotten.

Overall trust score. The overall trust score of a candidate measurement is a linear combination of the corresponding spatial trust score and temporal trust score. Specifically, we define the *trust score* T_j of candidate measurement R'_j as

$$T_j = \omega T_j^s + (1-\omega)T_j^t,$$

where $\omega \in [0,1]$ is another system parameter indicating the weight given to the spatial trust score.

2.4.3.4 Final REM Construction

After the iterative selection process terminates, the DBA constructs a final REM using the trusted measurements $\{R'_j | j \in \Theta_t\}$. In particular, the DBA refits the empirical semivarogram model using $\{R'_j | j \in \Theta_t\}$ as in the evaluation of spatial trust scores. For every cell center $\mathbf{x}_c, c \in \{1,\ldots,N\}$, the DBA predicts it residue RSS $\hat{S}(\mathbf{x}_c)$ using Eq. (2.2) and outputs its estimated RSS as

$$\hat{Z}(\mathbf{x}_c) = \hat{S}(\mathbf{x}_c) + P(\mathbf{x}_c) ,$$

where $P(\mathbf{x}_c)$ is the predicted linear trend.

2.4.4 Discussion

As mentioned before, the DBA terminates the process if the ratio between the number of the trusted measurements and the total number of measurements reaches a predetermined threshold η. This terminal condition assumes that the ratio of false measurements is small, and the DBA intends to defend against up to $1 - \eta$ ratio of false measurements.

There are another two possible terminal conditions with each corresponding to a different assumption about the attacker. First, the iterative measurement selection process may terminate when the number of trusted measurements reaches a predefined threshold, *i.e.*$|\Theta_t| \geq \eta_2$, where $\eta_2 \in [|\Theta_a|, |\Theta_a \bigcup \Theta_m|]$ is a system parameter. This terminal condition assumes that there are sufficient good measurements, while the ratio of the number of false measurements over the total number of measurements could be potentially large. Using this terminal condition, the DBA intends to construct an REM with sufficiently high accuracy with just enough trusted measurements even if there are additional good measurements that can be explored. Second, the iterative measurement selection process may terminate when no remaining candidate measurement has a trust score exceeding η_3, where $\eta_3 \in [0, 1]$ is a system parameter. This terminal condition assumes that false measurements exhibit high inconsistency in comparison with trusted measurements, i.e., with large T_j. Note that under this terminal condition, the last iteration may add fewer than q candidate sensors to the trust sensor set.

2.5 Performance Evaluation

In this section, we firstly introduce the spectrum measurement dataset used for evaluation and the detrending procedure that we use. We then report our simulation results.

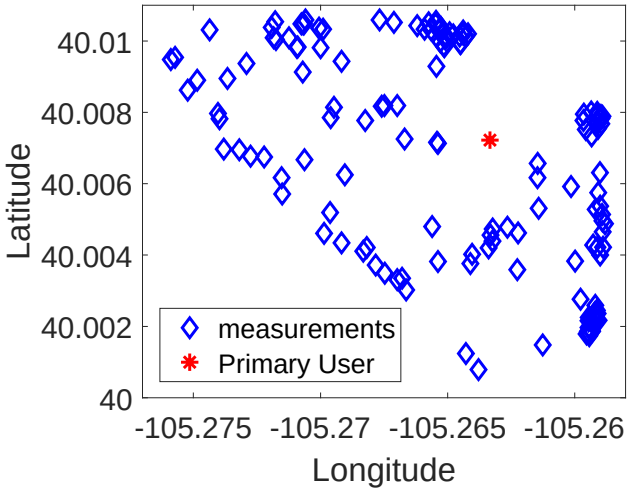


Figure 2.2: The locations of measurements and the PU in `cu/wimax` dataset.

2.5.1 Dataset

I use the CRAWDAD `cu/wimax` dataset [59] for the simulation studies, which was also used in [29]. The `cu/wimax` dataset was collected at the University of Colorado Boulder (UC) and contains the Carrier to Interference plus Noise Ratio (CINR) measurements of the WiMax network consisting of 5 base stations serving the UC campus taken by a portable spectrum analyzer. The measurements were taken on a 100m equilateral triangular lattice and additional measurements taken at random and optimized points. In our simulation studies, we choose the measurements for channel 308 and BSID 3674210305, which includes 145 measurements at different locations. Fig. 2.2 shows the locations of the measurements and the PU.

2.5.2 Measurement Detrending

We follow the detrending procedure in [29] to remove the potential source of non-linear trend from the measurements. Specifically, for each CINR measurement $Z_{\text{cinr}}(\mathbf{x})$ at location $\mathbf{x}$, we first convert it into the corresponding path loss value by computing

$$Z_{\text{pl}}(\mathbf{x}) = \mathcal{T} + G_{\text{tx}} - N - Z_{\text{cinr}}(\mathbf{x}) \ ,$$

Table 2.1: Default Simulation Settings

Para.	**Val.**	**Description.**
$\lvert\Theta_t\rvert$	10	The number of trusted measurements
$\lvert\Theta_c\rvert$	90	The number of candidate measurements
ω	0.5	Weight of spatial trust score
	20	The number of false measurements
T	5 dB	Attack strength
q	10	Step length
η	80	Terminal condition 1
η_1	0.8	Terminal condition 2
η_2	0.8	Terminal condition 3

where $\mathcal{T} = 40$dBm is the PU's transmission power, G_{tx} =10dB is the receiver antenna gain, and $N = -95$dBm is the constant noise floor value. Second, we compute the predicted pass loss using an empirical log-distance path loss model as

$$P(\mathbf{x}) = \alpha 10 \log_{10}(d) + 20 \log_{10}(f) + 32.45 + \epsilon \,, \tag{2.6}$$

where d is the distance between $\mathbf{x}$ and the PU, $f = 2578$MHz is the PU's transmitting frequency, 32.45 (dB) represents the free-space path loss, $\alpha = 1.22$ and $\epsilon = 28.81$dB are the path loss exponent and the offset obtained by fitting the measurements. The detrended measurement is then given by

$$S(\mathbf{x}) = Z_{\text{pl}}(\mathbf{x}) - P(\mathbf{x}). \tag{2.7}$$

2.5.3 Simulation Settings

We divide the 145 measurements into two sets: a testing set $\mathcal{R}_t$ with 100 measurements and a validating set $\mathcal{R}_v$ with 45 measurements as the ground truth. From the 100 testing measurements, we randomly choose 10 measurements as trusted ones and another 20 measurements as the false ones. Moreover, we define a false measurement R_i has an *attack strength* T (dB) if it reports a $Z_i + T$ where Z_i is the true measurement [16]. Table 1 summarizes our default simulation settings unless mentioned otherwise.

We primarily use Mean Absolute Error (MAE) to evaluate the performance of ST–REM. In particular, for each measurement $R_i \in \mathcal{R}_v$, let Z_i and $\hat{Z}_i$ be the reported RSS and estimated RSS, respectively. The MAE is defined as

$$\text{MAE} = \frac{\sum_{R_i \in \mathcal{R}_v} |Z_i - \hat{Z}_i|}{|\mathcal{R}_v|}. \tag{2.8}$$

Since ST–REM is the first solution for secure REM construction against false spectrum measurements, we compare its performance with the following three strategies.

- **Trusted measurements only (TMO)**: the DBA constructs the REM using the measurements submitted by anchor sensors only.
- **All measurements (AM)**: the DBA constructs the REM constructed using all measurements, including false ones.
- **All but false measurements (ABFM)**: the DBA constructs the REM constructed using all the measurements except for the false ones. Note that since the DBA does not know which measurements are false in reality, the accuracy achieved under ABFM can be viewed as the upper bound of any mechanism that can achieve.

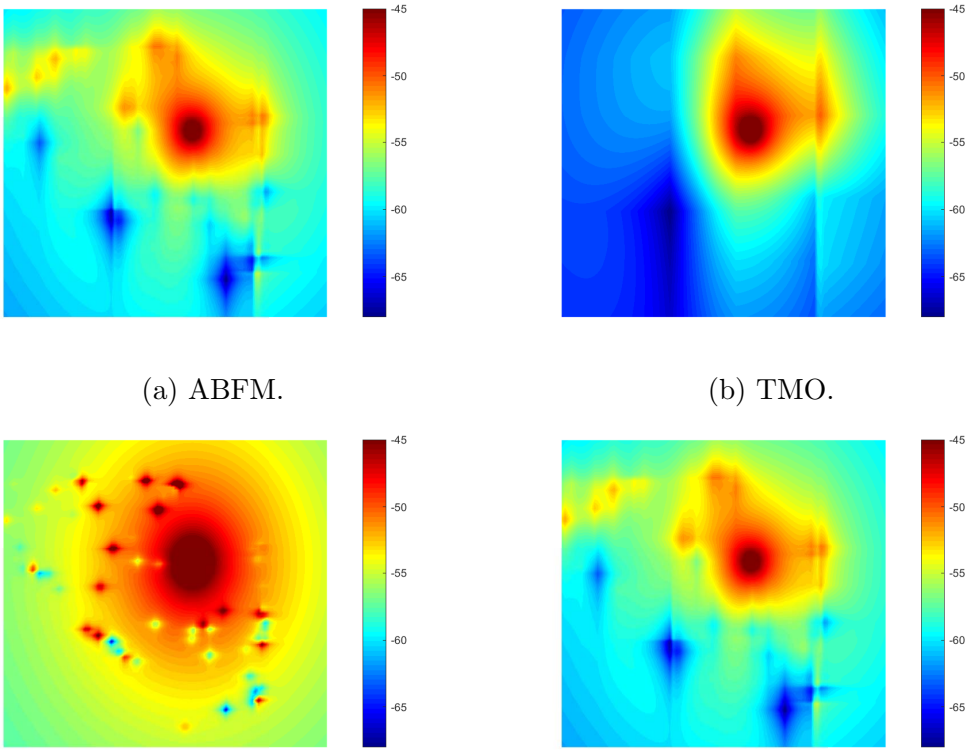
2.5.4 Simulation Results

We now report the simulation results for comparison of TMO, AM, ABFM, and ST–REM.

2.5.4.1 Exemplary REMs Constructed by TMO, AM, ABFM, and ST–REM

Fig. 2.3 shows four exemplary REMs constructed by ABFM, TMO, AM, and ST–REM, respectively, where attack strength T is 5dB. Each REM is constructed by estimating the path loss value at the center of every cell and then converting the predicted path loss value back into RSS by computing

$$\hat{Z}(\mathbf{x}) = \mathcal{T} + G_{\text{tx}} - (\hat{S}(\mathbf{x}) + P(\mathbf{x})).$$



(a) ABFM. (b) TMO.

(c) AM. (d) ST–REM.

Figure 2.3: Exemplary REMs constructed by TMO, AM, ABFM, and ST–REM with 10 trusted and 20 false measurements.

Specifically, Fig. 2.3a shows the REM constructed by ABFM using all the good measurements, which can serve as the baseline for other mechanisms. Generally speaking, the closer the REM to the REM constructed by ABFM, the more resilient the mechanism against false spectrum measurements. Fig. 2.3b shows the REM constructed using only the 10 known trusted measurements from anchor sensors, which is very coarse and different from the REM constructed by ABFM. This shows that the REM constructed using only a small number of known trusted measurements is very coarse. On the other hand, Fig. 2.3c shows that the REM constructed using all the measurements is highly distorted by the 20 false measurements, which highlights the detrimental impact of even a small number of false measurements. Finally, Fig. 2.3d shows the REM constructed by ST–REM. As we can see, the REM is very close to the REM constructed by ABFM

shown in Fig. 2.3a, indicating the high resilience of ST–REM to false measurements. These exemplary REMs demonstrate that the significant advantage of ST–REM over both TMO and AM.

2.5.4.2 Impact of Attack Strength T

Fig. 2.4 shows the MAEs under ABFM, TMO, AM, and ST–REM with the attack strength T varying from 0dB to 30dB. The MAEs under TMO and ABFM are not affected by the change in the attack strength and are plotted for reference only. As we can see, the MAE under ABFM, i.e., the ideal case, is approximately 2.67 dB. This represents the lower bound of the MAE of the REM constructed using Ordinary Kriging and coincides with the results obtained in the recent measurement study [28]. In addition, the MAE under TMO is around 4.86 dB, which again shows that the REM constructed from only a small number of trusted measurements is highly inaccurate. Moreover, the MAE under AM increases nearly linearly as the attack strength increases. In contrast, the MAE of ST–REM is very close to that of ABFM, which demonstrates the resilience of ST–REM against the change in attack strength.

2.5.4.3 Impact of the Number of False Measurements

Fig. 2.5 shows the MAEs under TMO, AM, and ST–REM with the number of false measurements varying from 0 to 50, where the MAE under TMO stays at

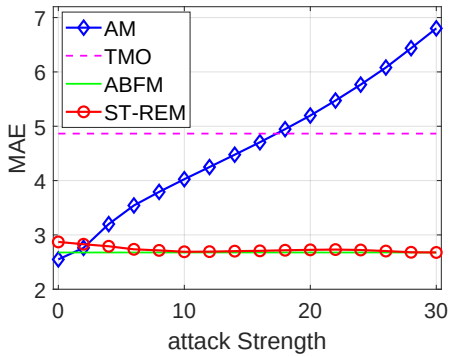


Figure 2.4: MAE vs. attack strength.

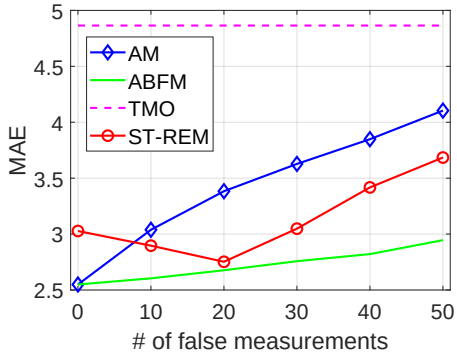


Figure 2.5: MAE vs. # of false measurements.

4.86 dB and is plotted for reference only. We can see that the MAE under AM is the same as that under ABFM when there is no false measurement and increases nearly linearly as the number of false measurements increases. This is anticipated, as the adverse impact of false measurements on the MAE grows as the number of false measurements increases. On the other hand, the MAE under ABFM slightly increases as the number of false measurements increases, which is caused by the corresponding decrease in the number of good measurements. In addition, the MAE under ST–REM initially declines as the number of false measurements increases. The reason for the initial decline is that ST–REM may terminate too early when there are only few false measurements, i.e., some good measurements are excluded from being used to improve the accuracy of the REM. As the number of false measurements approaches 20, fewer good measurements are discarded, and the MAE under ST–REM approaches that under ABFM. As the number of false measurements further increases from 20, the MAE under ST–REM deteriorates but is still much lower than that under AM. This is also expected, as ST–REM would include some false measurements in the final REM under such situations.

2.5.4.4 Impact of the Number of Trusted Measurements.

Fig. 2.6 compares the MAEs under ABFM, AM, and ST–REM with the number of trusted measurements, i.e., anchor sensors, varying from 10 to 80, where the MAEs under AM and ABFM are not affected and are plotted for reference only. As we can see, the MAEs under AM and ABFM are 3.38dB and 2.67dB, respectively. In addition, the MAE under TMO decreases from 4.86dB to 2.67dB as the number of trusted measurements increases from 10 to 80. This is anticipated, as the more good measurements, the higher the accuracy of the resulting REM, and vice versa. Moreover, while we can see that the MAE under ST–REM decreases as the number of trusted measurements increases, the gain resulted from additional trusted measurements is quite small. For example, the MAE under ST–REM is 2.76dB with 10 trusted measurements and decreases to 2.73dB with additional 10 trusted measurements. These results indicate

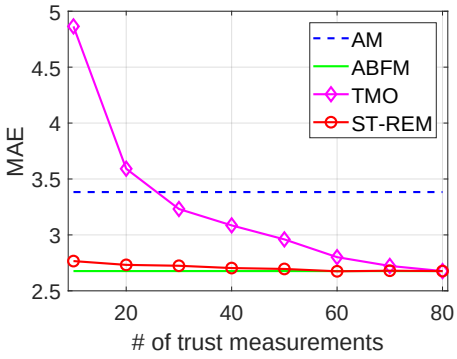


Figure 2.6: MAE vs. # of trust measurements.

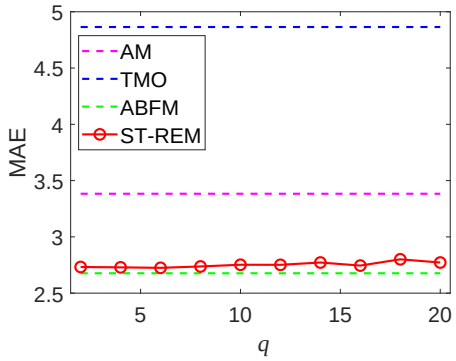


Figure 2.7: MAE vs. step length q.

that ST–REM only requires a small number of trusted measurements to achieve high accuracy of resulting REM.

2.5.4.5 Impact of Step Length q

Fig. 2.7 shows the MAEs under ST–REM with step length q varying from 2 to 20, where the MAEs under AM, TMO, and ABFM are not affected by the change in step length and are plotted for reference only. As we can see, the MAE under ST–REM slightly increases as the step length increases at the beginning. The reason is that the initial REM constructed from the measurements submitted by anchor sensors is quite coarse, and using the initial REM to estimate the trustworthiness of other measurements and add too many other measurements at once may have some false measurements included. This would lead to higher MAE of the final REM. As the step length further increases from 15 to 20, the MAE of the final REM slightly fluctuates. Overall, the change in step length has very limited impact on the accuracy of resulting REMs under the default settings.

2.5.4.6 Impact of Anchor Sensor Placement

We also evaluate the impact of anchor sensors' placement. Specifically, we consider the following four strategies for placing anchor sensors.

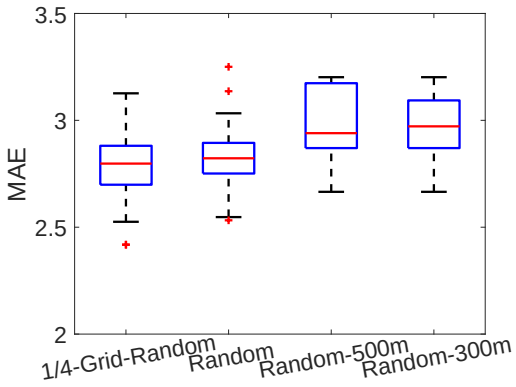


Figure 2.8: MAE vs. anchor sensor placement.

- *1/4–Grid–Random*: Divide the whole area into four square grids of equal size and randomly select 2 or 3 measurements in each grid to form the 10 trusted measurements.
- *Random*: Randomly select 10 measurements in the whole area as the trusted measurements.
- *Random-500m*: Randomly select 10 measurements within 500 meters of the PU as the trusted measurements.
- *Random-300m*: Randomly select 10 measurements within 300 meters of the PU as the trusted measurements.

Generally speaking, anchor sensors are distributed most evenly under 1/4-Grid-Random, followed by Random, Random-500m, and Random-300m.

Fig. 2.8 compares the MAEs under the four anchor sensor placement strategies for ST–REM. The median MAEs under 1/4-Grid-Random, Random, Random-500m, and Random-300m over 100 runs are 2.79dB, 2.82dB, 2.94dB, and 2.97dB, respectively. Generally speaking, the more unevenly anchor sensors are distributed, the higher the MAE, and vice versa. However, the difference among the four placement strategies are relatively small. Given the limited size of our dataset, we leave the further investigation of the optimal anchor sensor placement as our future work.

2.5.4.7 Comparison of SSO, TSO, and ST–REM.

Since ST–REM relies on both spatial and temporal trust scores to rank and select candidate measurements, we also compare it with the following two variants to better understand their effectiveness.

- *Spatial trust score only (SSO)*: The spatial trust score in ST–REM is given an weight of one, i.e., $\alpha = 1$ in Eq. (2.5).
- *Temporal trust score only (TSO)*: The spatial trust score in ST–REM is given zero weight, i.e., $\alpha = 0$ in Eq. (2.5).

Fig. 2.9 shows the Cumulative Distribution Functions (CDFs) of the MAEs under SSO under different attack strengths across 100 runs, where the CDF of ABFM is plotted for reference. As we can see, the MAE under SSO decreases as the attack strength increases. In particular, when the attack strength is 15dB, 94% of MAEs are higher than 3dB. In contrast, when the attack strength is 10dB and 5dB, the percentage drops to 87% and 31%, respectively. This is due to the fact that when the attack strength is small, e.g., 5dB, the differences between false measurements and good measurements are quite small, making it difficult to differentiate them and resulting in a relatively high MAE. It also indicates that SSO is most effective if the attack strength is large.

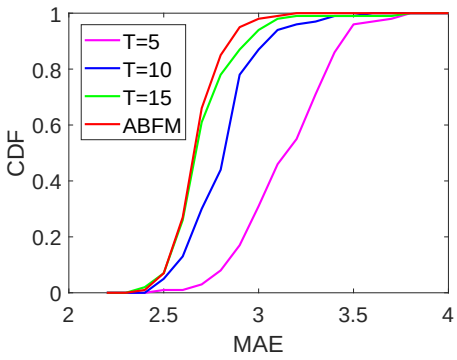


Figure 2.9: CDF of MAE under SSO.

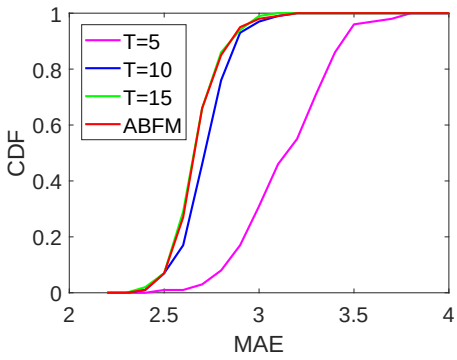


Figure 2.10: CDF of MAE under TSO in fifth epoch.

Fig. 2.10 shows the CDFs of the MAEs under TSO under different attack strengths, where the CDF under ABFM is plotted for reference only. We can see that when the attack strength keeps 5dB in the previous four epochs, the MAE under TSO in the fifth epoch is much higher than that under ABFM. In contrast, when the attack strength is 15dB in the previous four epochs, the CDF of the MAEs under TSO matches closely with that of ABFM in the fifth epoch. This is anticipated, because the larger the attack strength, the later a false measurement is added into the trusted measurement set, the higher the temporal trust score of the false measurement, and vice versa. It is thus easier for TSO to differentiate false measurements from good ones when the attack strength is high.

2.5.4.8 Impact of Sudden Change in Attack Strength

To evaluate the effectiveness of spatial and temporal trust scores in filtering out false measurements in the presence of sudden change in the attack strength, we further consider the following two exemplary attack strategies.

- *Attack Strategy 1-sudden decrease in the attack strength*: The attacker chooses an attack strength of 15dB in the first four epochs and changes the attack strength to 5dB in the fifth epoch.
- *Attack Strategy 2-sudden increase in attack strength*: The attacker chooses an attack strength of 5dB in the first four epochs and changes the attack strength to 15dB in the fifth epoch.

Fig. 2.11 shows the CDFs of MAEs in the fifth epoch under ST–REM, SSO, TSO, and ABFM under Attack Strategy 1, where the CDF of the MAE under ABFM is plotted for reference only. We can see that the MAE under ST–REM is very close to that under TSO and much lower than that under SSO. In particular, the CDF of MAEs under ST–REM and TSO are close to the one under ABFM, while the CDF of the MAEs under SSO is quite far from that under ABFM. In addition, the CDF of the MAEs under TSO overlaps with the one under ABFM. The reason is that as the

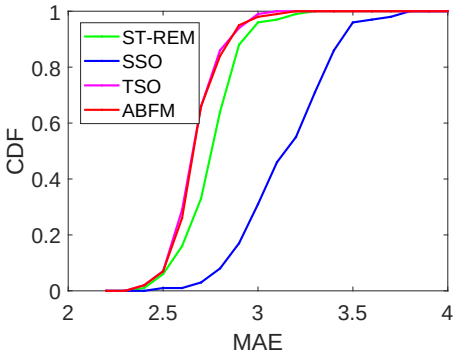


Figure 2.11: CDF of MAE under Attack Strategy 1.

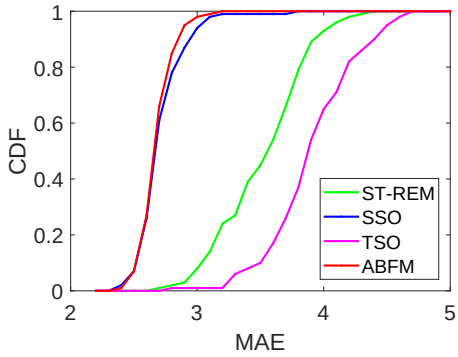


Figure 2.12: CDF of MAE under Attack Strategy 2.

attack strength in the previous epoch is relatively high, e.g., 15dB, false measurements are easier to be filtered out by SSO and ST–REM, which result in lower temporal trust scores for false measurements in the current epoch. In contrast, since the attack strength is relatively small, i.e., 5dB, in the current epoch, the spatial trust score of false measurements are relatively small, making it difficult to filter out false measurements by SSO, leading to a higher MAE under SSO.Although SSO alone is less effective under Attack Strategy 1, ST–REM is still able to differentiate false measurements from good ones by jointly considering the temporal trust scores of the measurements.

Fig. 2.12 shows the CDFs of the MAEs under ST–REM, SSO, TSO, and ABFM under Attack Strategy 2, where the CDF of the MAE under ABFM is again plotted for reference. We can see that ST–REM outperforms TSO, but it is less effective than SSO. The reason is that under Attack Strategy 2, the attack strength in each previous epoch is 5dB, which is too small to always assign high temporal trust scores for false measurements. Thus, the CDF of MAEs under TSO is far from the CDF of MAEs under ABFM. In contrast, since the attack strength in current epoch is 15dB, which is large enough to filter out false measurements correctly, the CDF of MAEs under SSO is very close to the ideal case. In this circumstance, although the temporal trust score is not reliable, ST–REM is also powerful to exclude false measurements with the benefit of spatial trust score.

These results indicate that SSO is most effective in filtering out false measurements when the attack strength is high in the current epoch, while TSO can differentiate false measurements from good ones as long as the attack strength is high enough in previous epochs. By jointly considering the spatial and temporal trust scores, ST–REM can effectively filter out false measurements as long as the attacker chooses a high attack strength in any epoch.

2.5.4.9 Impact of Dynamic Attack Strength

We also evaluate the impact of dynamic attack strengths by considering the following three attack strategies: gradually ascending attack strengths, gradually descending attack strengths, and static attack strengths.

Fig. 2.13 shows the MAE under ST–TEM with the attack strength gradually increased from 0 by 2dB in each epoch for 15 epochs and different ωs, where the MAEs under ABFM is plotted for reference. We can see that the MAE under ST–REM initially increases and then gradually decreases until reaching the MAE under ABFM under all weight ωs. In addition, the higher the weight ω, the earlier the MAE under ST–REM starts to decrease, and thus the earlier converge to that under ABFM. The reason is that when the attack strength is small, e.g., 2dB in the second epoch, false measurements are very similar to good ones, and ST–REM is unable to filter

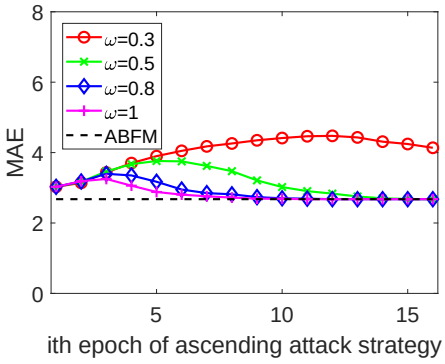


Figure 2.13: MAE under gradually ascending attack strength.

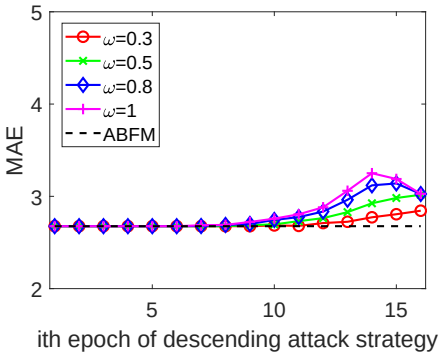


Figure 2.14: MAE under gradually descending attack strength.

out all false measurements. As the attack strength further increases, while false measurements become easier to filter out by ST–REM, some false measurements are still deemed trusted by ST–REM, and their overall impact on the MAE increases due to higher attack strength. As the attack strength keeps increasing, more and more false measurements are detected by ST–REM and excluded from the final REM, resulting in the overall decrease in the MAE under ST–REM. In addition, we can see that the higher the weight ω, the earlier the MAE starts to decrease, and vice versa. This is because spatial trust score is more effective than temporal trust score in filtering out false measurements with increasing attack strength.

Fig. 2.14 shows the MAE under ST–TEM with the attack strength gradually decreased from 30dB by 2dB in each epoch for 15 epochs and different ωs, where again the MAEs under ABFM is plotted for reference. We can see that the MAE under ST–REM is the same as that under ABFM for the first eight epochs for all ωs. This is because false measurements with large attack strength, e.g., 16dB in the eighth epoch, are very different from good ones and can be easily filtered out by ST–REM. As the attack strength further decreases, the MAE under ST–TEM first increases and then decreases under different ωs. The reason is that as the attack strength becomes smaller, some false measurements will be deemed trusted under ST–REM, leading to the increase in the MAE. As the attack strength keeps decreasing, while more false measurements will be added to the trusted measurement set under ST–REM, their accumulative impact on the MAE becomes smaller. Moreover, we can see that the higher the weight ω, the larger the maximum MAE the attacker can achieve over the 16 epochs. This is because the spatial trust score alone is less effective in filtering out false measurements with small attack strengths and the smaller the weight given temporal trust score, the less likely a false measurement can be filtered out by ST–REM.

Fig. 2.15 shows the average temporal trust score of good and false measurements over 15 epochs, where the attack strength stays at 5dB, 10dB, and 15dB for all epochs. We can see that the average temporal trust score of good measurements decreases

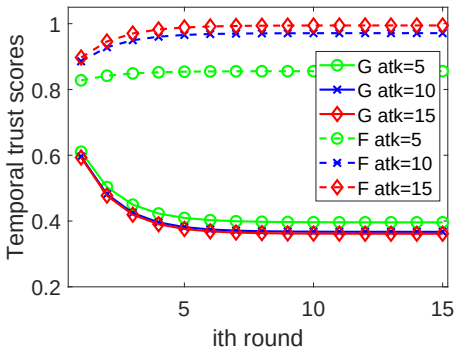


Figure 2.15: Temporal trust scores multiple epochs under equal attack strategies.

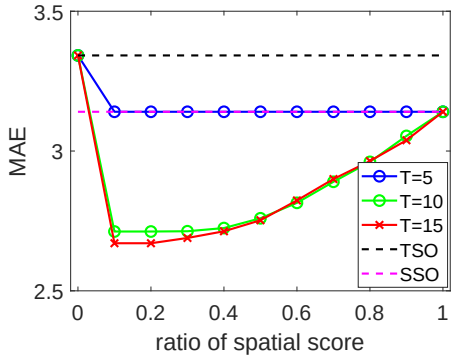


Figure 2.16: MAE vs. weight ω.

rapidly in the first few epochs and then remains stable in the following epochs. In contrast, the average temporal trust score of false measurements increases in the first few epochs and becomes stable in the following epochs. The reason that the average temporal trust scores of good and false measurements change slower in later epochs is as follows. In the first epoch, all the measurements are assigned the same initial temporal score η, and measurements are added to the trusted measurement set entirely based on their spatial trust scores. The order in which the measurements are added to the trusted measurement set results in the update in their temporal trust scores. In each of the subsequent epochs, false measurements with higher temporal trust scores will have higher overall trust scores and thus be added to the trusted measurement even later than in the previous epoch, if ever. This process results in the continuous decrease in the average temporal trust score of good measurements as well as the continuous increase in that of false ones. Finally, we can see that the higher the attack strength, the larger the gap between the average temporal trust score of good measurements and that of false ones, and vice versa, which is expected.

To conclude, ST–REM can achieve an acceptable performance as long as the spatial trust score is reliable or the temporal trust score is reliable. In addition, for the temporal trust score, it is reliable as long as there are several large enough attack strength appeared in the previous epochs.

2.5.4.10 Impact of the Weight ω.

Fig. 2.16 shows the MAE of ST–REM with the weight of spatial score ω varying from 0 to 1, where the MAEs under TSO ($\omega = 0$) and SSO ($\omega = 1$) are plotted for reference. Here we assume that the attack strength in current epoch is 5dB and that in the previous three epochs is 5dB, 10dB and 15dB, respectively. We can see that as ω increases from 0 to 0.1, the MAE under ST–REM first decreases sharply from 3.34dB under TSO to 3.14dB, 2.71dB and 2.67dB when attack strength in previous epoch is 5dB, 10dB, and 15dB, respectively. As ω further increases from 0.1 to 1, the MAE under ST–REM gradually increases to 3.14dB achieved by SSO under all three attack strengths in previous epochs. This result shows that there is always an optimal weight ω assignment under which ST–REM outperforms both SSO and TSO.

2.6 Summary

In this chapter, we have introduced the design and evaluation of ST–REM, a novel spatiotemporal approach for securing crowdsourced REM construction in the presence of false spectrum measurements. Inspired by self-labeled techniques, ST–REM gradually constructs an initial REM from trusted measurements from a small number of anchor sensors and then iteratively refine it by gradually incorporating the measurements from mobile sensors that are deemed most trustworthy. The trustworthiness evaluation in ST–REM jointly considers a measurement's spatial fitness of trusted measurements and the long-term behavior of the mobile sensor. Extensive simulation studies using a real spectrum measurement dataset confirm that the proposed approach can produce an REM with sufficient accuracy in the presence of false measurements.

Chapter 3

DIFFERENTIALLY-PRIVATE INCENTIVE MECHANISM FOR CROWDSOURCED RADIO ENVIRONMENT MAP CONSTRUCTION

3.1 Introduction

Crowdsourcing-based REM construction requires sound incentive mechanisms to stimulate crowdsourcing workers' participation. In particular, performing spectrum sensing incurs non-trivial effort to crowdsourcing workers, such as their time and device battery. Without strong incentives, potential workers may be reluctant to participate in crowdsourcing-based spectrum sensing. A common approach for providing incentives in mobile crowdsourcing systems is to use reverse auction [60], where crowdsourcing workers sell their services by submitting their bids to the DBA, which in turn selects a subset of bidders as winners and offers payments based on their bids. Reverse auction has been widely used in many mobile crowdsourcing systems such as [61, 62].

A sound reverse auction mechanism for crowdsourcing-based REM construction needs to satisfy three critical requirements. First, crowdsourcing workers are selfish in reality and may lie about their costs if doing so can increase their utilities. This requires the reverse auction mechanism to be *truthful*, which means that bidding the true sensing cost is the optimal strategy for mobile crowdsourcing workers. Second, mobile crowdsourcing workers' bids may reveal their personal information, such as their locations [63, 64] and opportunity costs. While the DBA is commonly assumed to be trusted, curious workers could infer other workers' bids from the change in the payment profiles by submitting different bids for the same sensing task in different rounds [65]. It is thus necessary to protect crowdsourcing workers' bid privacy against other curious workers. Last but not least, reverse auction involves the selection of a set of winners,

which needs to ensure the accuracy of the resulting REM. However, the optimal selection of winners to maximize REM accuracy is an NP-hard problem even without considering the first two requirements. Despite the large body of work on privacy-preserving incentive mechanisms for mobile crowdsourcing systems [66, 67, 68, 69, 70], none of them satisfy the above three requirements. There is thus a pressing need to develop sound privacy-preserving incentive mechanisms to stimulate crowdsourcing workers' participation while protecting their bid privacy and ensuring high REM accuracy.

In this chapter, we tackle this challenge by introducing DPS, a novel differentially-private reverse auction mechanism which can simultaneously ensure bid privacy for crowdsourcing workers and the accuracy of the constructed REM. In DPS, every crowdsourcing worker submits a bid for performing spectrum sensing at his current location. Serving as the auctioneer, the DBA selects a subset of workers as winners based on the received bids and determines the payment to the winners. The key ingredient of DPS is a greedy algorithm for selecting a candidate winner set with guaranteed REM accuracy with respect to every possible payment price and choosing the final winner set with corresponding payment price using the exponential mechanism to ensure differential privacy for individual workers. Our main contributions can be summarized as follows.

- To the best of our knowledge, we are the first to study differentially-private mechanism design for crowdsourcing-based REM construction.
- We introduce a novel differentially-private reverse auction mechanism that can simultaneously provide differential privacy to crowdsourcing workers' bids, approximate truthfulness, and guaranteed REM accuracy at the DBA.
- We thoroughly evaluate the proposed mechanism via a combination of theoretical analysis and detailed simulations studies using real spectrum measurement data, which confirm the efficacy and efficiency of the proposed mechanism.

The rest of this chapter is structured as follows. Section 3.2 discusses the related work. Section 3.3 introduces the necessary background of statistical interpolation technique and our system model along with design goals. Section 3.4 introduces the design

of our solution. Section 3.5 analyzes the performance of DPS. Section 3.6 reports the simulation results, and Section 3.7 concludes this work.

3.2 Related Work

In this section, we discuss some of the prior work in several areas related to our work.

Differentially-private mechanism design has attracted many attentions in recent years. McSherry and Talwar [60] introduced the first differentially private auction mechanism by incorporating the exponential mechanism. General methods for designing auction mechanisms with differential privacy guarantee were studied in [66, 71, 72, 67]. All these works focus on maximizing social welfare and are inapplicable to crowdsourced REM construction where the objective is to maximize the average K-var reduction.

Several privacy-preserving mechanisms have been proposed for spectrum allocation problem. THEMIS [73] incorporated cryptographic technique into spectrum auction to deal with the seller-side fraudulent actions. Huang et al. [69] proposed a truthful and privacy-preserving mechanism to achieve k-anonymity in spectrum auctions. Subsequently, PPS [68] applied homomorphic encryption to maximizes the social efficiency and preserves bid privacy. All these solutions rely on cryptographic techniques and incur high computation and communication overheads. Moreover, neither of them provide differential privacy guarantee for individual worker's bid.

DEAR [74] integrated the exponential mechanism with spectrum auction to achieve approximate truthfulness, privacy preservation, and approximate revenue maximization. Zhu *et al.* [75] also incorporated differential privacy to design a truthful auction mechanism for dynamic spectrum redistribution. BidGuard [65] is a differentially private auction mechanism aiming at minimizing social cost. Jin *et al.* [76] designed a differentially-private incentive mechanism to protect workers' bid privacy against honest-but-curious workers, and a total payment minimization problem is formulated to ensure the truthfulness and workers' utility. However, these solutions assume that

auctioneers have a prior knowledge of the bidders' valuation distribution and focus on the revenue maximization. They are thus not directly applicable to our context.

Another line of research is to design truthful auction mechanisms for mobile crowdsourcing systems. Yang *et al.* [61] designed an incentive mechanism with the objective function maximizing platform utility to satisfy individual rationality and truthfulness. Zhao *et al.* [62] also aimed at selecting a subset of users to maximize the value of service from selected mobile users subjected to individual rationality. TRAC [77] is a truthful auction mechanism for location-aware crowdsensing systems. Ying *et al.* [13] introduced an incentive mechanism for crowdsourcing-based spectrum sensing, which achieved approximate maximization of K-var reduction by considering crowdsourcing workers' marginal contribution. None of these solutions consider users' bid privacy and thus cannot be applied to our target problem. A truthful reverse auction is introduced in [78] for crowdsourcing-based data aggregation, which provides differential privacy for sensed data. Truthful double auction has also been studied in [79] for crowdsourcing systems involving multiple auctioneers. None of them considered protecting crowdsourcing workers' bid privacy.

3.3 Preliminaries

In this section, we introduce the system and adversarial models, crowdsourcing-based REM construction, the auction model, and our design objectives.

3.3.1 System Model

We consider a DBA which maintains an REM for the spectrum availability in its service area $\mathcal{D} \in \mathbb{R}^2$. The area $\mathcal{D}$ is divided into a number of cells of equal size.

The DBA relies on spectrum sensing to constructs and maintains the REM. Specifically, the DBA deploys a small number of static spectrum sensors at strategic locations and outsources the majority of spectrum sensing tasks to mobile crowdsourcing workers. Deploying few static spectrum sensors cannot only guarantee minimum level of service when there are insufficient mobile crowdsourcing workers, e.g., during

nighttime, but also facilitate detection of potential false spectrum measurements [80]. Denote by $\mathcal{S}$ the set of dedicated spectrum sensors and $\mathcal{N} = \{1, \ldots, n\}$ the set of crowdsourcing workers. We assume that the locations of dedicated spectrum sensors are known to the DBA. We also assume that each crowdsourcing worker owns a mobile device capable of spectrum sensing and acquiring its current location.

The DBA periodically collects spectrum measurements from both static spectrum sensors and selected crowdsourcing workers to update the REM. Assume that the time is divided into epochs. At the beginning of each epoch, the DBA broadcasts a spectrum sensing request to all the potential crowdsourcing workers in $\mathcal{D}$, which includes sensing frequency, sampling rate, etc. On receiving the sensing request, each crowdsourcing worker $i \in \mathcal{N}$ submits a bid b_i along with his location $\mathbf{x}_i$ to the DBA, indicating that he is willing to perform spectrum sensing at location $\mathbf{x}_i$ for a minimal payment of b_i. Once the DBA receives a bid-location profile $(b, \mathcal{X})$ where $b = (b_1, \cdots, b_n)$ and $\mathcal{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)$, it selects a winner set $\mathcal{W} \subseteq \mathcal{N}$ and determines the payment p_i for each winner $i \in \mathcal{W}$.

The DBA then informs the winners and collects spectrum measurements from them as well as static spectrum sensors. In particular, each static sensor or winning crowdsourcing worker $i \in \mathcal{S} \bigcup \mathcal{W}$ submits a spectrum measurement $Z(\mathbf{x}_i)$ to the DBA. On receiving all the measurements $\{Z(\mathbf{x}_i) | i \in \mathcal{S} \bigcup \mathcal{W}\}$, the DBA estimates the RSS at the center of every cell using Eq. (3.1) whereby to produce the updated REM.

3.3.2 The Objective Function at the DBA

A primary goal of the DBA is to maximize REM accuracy, for which Kriging Variance reduction has been proposed as a proper metric.

Recall that under Ordinary Kriging (OK) [54], the RSS at an unmeasured location $\mathbf{x}_0$ is estimated from the RSSs at measured locations. Specifically, given a set of spectrum measurements at locations $\mathcal{X} = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, the RSS at location $\mathbf{x}_0$ is estimated as

$$\hat{Z}(\mathbf{x}_0) = \sum_{i=1}^{n} \omega_i Z(\mathbf{x}_i), \tag{3.1}$$

where $\sum_{i=1}^{n} \omega_i = 1$ are normalized weights. It is easy to see that $\hat{Z}(\mathbf{x}_0)$ is a linear unbiased estimator as $\mathbb{E}(\hat{Z}(\mathbf{x}_0) - Z(\mathbf{x}_0)) = \mathbb{E}(\sum_{i=1}^{n} \omega_i Z(\mathbf{x}_i) - Z(\mathbf{x}_0)) = \sum_{i=1}^{n} \omega_i \mathbb{E}(Z(\mathbf{x}_i)) - \mathbb{E}(Z(\mathbf{x}_i)) = \mu \sum_{i=1}^{n} \omega_i - \mu = 0$.

By minimizing the Mean Squared Error (MSE) $\mathbb{E}[(\hat{Z}(\mathbf{x}_0) - Z(\mathbf{x}_0))^2]$ with respect to $\{\omega_i\}$ under the normalization constraint $\sum_{i=1}^{n} \omega_i = 1$, we can obtain a set of linear equations, commonly referred to as Kriging system.

Solving the Kriging system leads to the optimal coefficients given by

$$\omega^* = (\omega_i^*)_{i \in \mathcal{X}} = \Sigma_{\mathcal{X}\mathcal{X}}^{-1} \Sigma_{\mathcal{X}\mathbf{x}_0}, \tag{3.2}$$

where $\Sigma_{\mathcal{X}\mathcal{X}}^{-1}$ is the covariance matrix, and $\sum_{\mathcal{X}\mathbf{x}_0}$ is the vector of cross-covariances between every $Z(\mathbf{x}_i)(i \in [1, n])$ and $Z(\mathbf{x}_0)$. Since the estimator is unbiased, the minimized MSE, commonly referred to as *Kriging variance (K-var)*, is given by

$$\sigma_{\mathbf{x}_0|\mathcal{X}}^2 = \sigma_{\mathbf{x}_0}^2 - \Sigma_{\mathcal{X}\mathbf{x}_0}^T (\Sigma_{\mathcal{X}\mathcal{X}}^{-1}) \Sigma_{\mathcal{X}\mathbf{x}_0} \ ,$$

where $\sigma_{\mathbf{x}_0}^2$ is the unknown K-var when $\mathcal{X} = \emptyset$. K-var represents the prediction uncertainty at the unmeasured location and is often used as the estimator design metric. The smaller K-var, the higher accuracy of the estimation, and vice versa.

The DBA's primary objective is to choose the set of winners $\mathcal{W}$ with total payment under the budget constraint while minimizing the average K-var of the produced REM over its service region.

We adopt an objective function similar to [13], where The DBA chooses winners partially based on the predicted contribution of additional measurements submitted at the winners' locations. Specifically, under the optimal weights given in Eq.(3.2), the K-var at an unmeasured location $\mathbf{x} \in \mathcal{D}$ after taking measurements from deployed dedicated sensors $\mathcal{S}$ at locations $\mathcal{X}_\mathcal{S} = \{\mathbf{x}_i | i \in \mathcal{S}\}$ is given by [13].

$$\sigma_{\mathbf{x}|\mathcal{X}_\mathcal{S}}^2 = \sigma_{\mathbf{x}}^2 - \Sigma_{\mathcal{X}_\mathcal{S}\mathbf{x}}^T \Sigma_{\mathcal{X}_\mathcal{S}\mathcal{X}_\mathcal{S}}^{-1} \Sigma_{\mathcal{X}_\mathcal{S}\mathbf{x}}, \tag{3.3}$$

where $\sigma_{\mathbf{x}}^2$ is the unknown variance at location $\mathbf{x}$, $\Sigma_{\mathcal{X}_\mathcal{S}\mathcal{X}_\mathcal{S}}$ is the covariance matrix of all measurements from dedicated sensors, and $\Sigma_{\mathcal{X}_\mathcal{S}\mathbf{x}}$ is the vector of cross-covariances between $\{Z(\mathbf{x}_i) | i \in \mathcal{S}\}$ and $Z(\mathbf{x})$.

Given a winner set $\mathcal{W}$, the DBA will collect additional spectrum measurements from locations $\mathcal{X}_{\mathcal{W}} = \{\mathbf{x}_i | i \in \mathcal{W}\}$. Combing spectrum measurements from $\mathcal{S}$ and $\mathcal{W}$, the Kriging variance at an unmeasured location $\mathbf{x} \in \mathcal{D}$ is given by

$$\sigma^2_{\mathbf{x}|\mathcal{X}_{\mathcal{S}\cup\mathcal{W}}} = \sigma^2_{\mathbf{x}} - \Sigma^T_{\mathcal{X}_{\mathcal{S}\cup\mathcal{W}}\mathbf{x}} \Sigma^{-1}_{\mathcal{X}_{\mathcal{S}\cup\mathcal{W}}\mathcal{X}_{\mathcal{S}\cup\mathcal{W}}} \Sigma_{\mathcal{X}_{\mathcal{S}\cup\mathcal{W}}\mathbf{x}}. \tag{3.4}$$

Subtracting Eq. (3.4) from Eq. (3.3), we can obtain the predicted Kriging variance reduction at location $\mathbf{x}$ caused by additional measurements from $\mathcal{W}$ as

$$\triangle\sigma^2_{\mathbf{x}}(\mathcal{W}) = \Sigma^T_{\mathcal{X}_{\mathcal{S}\cup\mathcal{W}}\mathbf{x}} \Sigma^{-1}_{\mathcal{X}_{\mathcal{S}\cup\mathcal{W}}\mathcal{X}_{\mathcal{S}\cup\mathcal{W}}} \Sigma_{\mathcal{X}_{\mathcal{S}\cup\mathcal{W}}\mathbf{x}} - \Sigma^T_{\mathcal{X}_{\mathcal{S}}\mathbf{x}} \Sigma^{-1}_{\mathcal{X}_{\mathcal{S}}\mathcal{X}_{\mathcal{S}}} \Sigma_{\mathcal{X}_{\mathcal{S}}\mathbf{x}}. \tag{3.5}$$

Now consider the whole service region $\mathcal{D}$. The average reduction of Kriging variance caused by the measurements submitted by winner set $\mathcal{W}$ is given by

$$f(\mathcal{W}) = \frac{1}{|\mathcal{D}|} \sum_{\mathbf{x}\in\mathcal{D}} \triangle\sigma^2_{\mathbf{x}}(\mathcal{W}). \tag{3.6}$$

Assume that the DBA has a budget B for payment to the winners for each epoch. The DBA intends to find a set of winners $\mathcal{W}$ along with payment profile $\{p_i | i \in \mathcal{W}\}$ under the budget constraint that maximizes the average reduction of Kriging variance in the service region $\mathcal{D}$, which can be formulated as the following optimization problem.

$$\begin{aligned} \text{Maximize} \quad & f(\mathcal{W}) \\ \text{subject to} \quad & \sum_{i\in\mathcal{W}} p_i \leq B, \\ & \mathcal{W} \subseteq \mathcal{N}. \end{aligned} \tag{3.7}$$

The above optimization problem is NP hard. In particular, let us temporally ignore the payment profile and budget constraints and assume that the DBA can choose a fixed number of winners. We can see that even this simplified version of the problem is a special case of subset selection problem, which is NP hard in general because of the non-linear nature of objective function $f(\mathcal{W})$.

3.3.3 Other Design Objectives

In addition to budget feasibility and maximizing the average K-var reduction in $\mathcal{D}$, we also intend to design our incentive mechanism to satisfy the following objectives.

Approximate truthfulness. Since crowdsourcing workers are selfish in reality, they submit bids different from their true valuations for the costs of performing spectrum sensing if doing so could increase their utilities. Assume that each crowdsourcing worker i has a true valuation v_i for the cost of performing spectrum sensing at location $\mathbf{x}_i$, which might be different from his bid b_i. The worker i's utility is then given by

$$u_i = \begin{cases} p_i - v_i, & \text{if } i \in \mathcal{W}, \\ 0, & \text{otherwise}, \end{cases} \tag{3.8}$$

where p_i is the payment worker i receives from the DBA if he is selected as a winner.

As a result, we aim to ensure that every crowdsourcing worker's optimal strategy is to bid his cost truthfully. Exact truthfulness, however, is usually difficult to achieve without losing other desirable properties. Instead, we aim to achieve γ-*truthfulness* such that no crowdsourcing worker can gain more than γ utility by bidding untruthfully.

Definition 1. *(γ-*truthful*). An auction mechanism is γ-truthful in expectation if and only if for any bid $b_i \neq v_i$ and any bid profile of other workers b_{-i},*

$$\mathbb{E}[u(v_i, b_{-i})] \geq \mathbb{E}[u(b_i, b_{-i})] - \gamma \ . \tag{3.9}$$

where γ is a small positive constant.

Differential privacy. We also intend to protect crowdsourcing workers' bidding privacy. While every worker's bid is known to the DBA and kept private from other workers, a curious worker could still infer other workers' bids by submitting different bids in different rounds of auction. Since the change in a single bid may result in significant change in the selected winner set and the payment profile, a curious worker may infer other workers' bids from the change in the payment he receives from the different payments she receives in different rounds. Differential privacy [81, 60] is a

powerful technique to protect bid privacy against such differential attacks. The key idea is that given two neighboring input datasets, a differentially-private mechanism will behave approximately the same on both datasets, which offers a strong guarantee that the presence or absence of a single element would not cause any major change in the output of the mechanism. The formal definition of differential privacy is given as follows.

Definition 2. *(**Differential privacy** [81, 60]). Let $M(\cdot)$ be a function that maps an input bid profile b to a payment profile $p \in P$. Mechanism $M(\cdot)$ is ϵ-differentially private if and only if for any set of payment profiles $R \subseteq P$ and any two bid profiles b and b' that differ in only one bid, we have*

$$\mathsf{Pr}[M(b) \in R] \geq \exp(\epsilon)\mathsf{Pr}[M(b') \in R] \ . \tag{3.10}$$

where ϵ is a small positive constant commonly referred to as privacy budget.

The exponential mechanism [60] is a classical tool to facilitate mechanism design via differential privacy. The key idea is to map a pair of input dataset A and candidate outcome o to a real valued "quality score" $q(A, o)$, where a higher score indicates better performance of the outcome. Given the output space $\mathcal{O}$, a score function $q()$, and the privacy budget ϵ, the exponential mechanism chooses the outcome $o \in \mathcal{O}$ with probability proportional to $\epsilon q(A, o)$.

Theorem 1. *[60] The exponential mechanism gives $2\epsilon\triangle$ differential privacy.*

Here $\triangle$ is the *global sensitivity* of $\epsilon q(A, o)$ that captures the largest change in the quality score by a single change of the input in A.

Computation efficiency. The selection of winner set and corresponding payment price should be computed in polynomial time.

Individual rationality. Our last design objective is individual rationality, which ensures that every crowdsourcing worker's utility is non-negative, i.e., $u_i \geq 0$ for all $i \in \mathcal{N}$, if he bids truthfully. The property is desired to stimulate mobile users' participation in any mobile crowdsourcing systems.

3.4 The DPS Design

In this section, we first give an overview of DPS and then detail its design.

3.4.1 Overview

DPS is designed by integrating a number of ideas. First, inspired by [74, 70] we adopt the single-price mechanism in which the DBA pays every winner the same amount of payment. It has been proved in [82] that the optimal single-price payment mechanism is within a constant factor of any differentiated payment mechanism. Second, under the single-price payment mechanism, we further design a greedy algorithm for selecting winners with guaranteed approximation ratio. Specifically, for any fixed payment price p, the maximum number of workers that the DBA can select is $\lfloor B/p \rfloor$. Any worker whose bid not higher than p can be chosen as a winner without violating the individual rationality. The winner selection problem under the single payment price p is then converted into the special case of subset selection problem which can be solved by greedy algorithm with guaranteed approximation ratio. Third, we choose final winner set and payment price using the exponential mechanism to ensure differential privacy. In particular, for each possible payment price, we can find a corresponding winner set and calculate the predicted average Kriging variance reduction. Given a set of possible payment prices, we then choose the final winner set and payment price using the exponential mechanism.

In what follows, we detail the DPS's design.

3.4.2 Detailed Design

We now detail the process of winner selection and payment price determination. On receiving the bid-location profile $(b, \mathcal{X})$, the DBA first finds a set of feasible payment prices. Without loss of generality, we assume that the possible payment to individual worker forms a finite set $P = \{p_{\min}, \ldots, p_{\max}\}$, where the lowest and highest payment prices are $p_{\min}$ and $p_{\max}$, respectively. Let $b_{\min}$ and $b_{\max}$ be the lowest and highest bids in b, respectively.

We say a price $p_k \in P$ is feasible if and only if there is at least one crowdsourcing worker with a bidding price no higher than p. The maximum number of winners is constrained by the budget B. In particular, given budget B and payment price p, the number of winners is at most $\lfloor B/p \rfloor$.

Second, for each feasible payment price $p_k \in P$, the DBA finds a winner set $\mathcal{W}_k$ using a greedy algorithm. The greedy algorithm explores the fact that the objective function $f(\cdot)$ in Eq. (3.12) is *submodular*, *non-negative*, and *monotone* [13]. Specifically, it is easy to see that the $f(\cdot)$ is non-negative as the K-var reduction is always positive for any non-empty winner set. Moreover, a set function $f : 2^{\mathcal{C}} \rightarrow \mathbb{R}$ is submodular if and only if $f(\mathcal{A} \bigcup \{x\}) - f(\mathcal{A}) \geq f(\mathcal{B} \bigcup \{x\}) - f(\mathcal{B})$ for any $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{C}$ and $x \in \mathcal{C} \setminus \mathcal{B}$. Submodularity captures the diminishing returns behavior of f: adding a new element to the input set always results in the increase in f, and the amount of increase reduces as the number of existing elements increases. Finally, $f(\cdot)$ is monotone if and only if $f(\mathcal{A}) \leq f(\mathcal{B})$ for any $\mathcal{A} \subseteq \mathcal{B} \subseteq \mathcal{C}$. A widely known result [83] is that for any function that is simultaneously submodular, monotone, and non-negative, a greedy algorithm that chooses the local optimal element at each step can find a solution with guaranteed approximation ratio of $1 - 1/e$, and no polynomial-time algorithm can achieve a better guarantee unless $P = NP$.

We now detail the greedy algorithm for winner selection for each payment price. Consider payment price p_k as an example, let $\mathcal{N}_k = \{i | b_i \leq p_k\}$ be the set of workers whose bids are not higher than p_k. The DBA maintains a winner set $\mathcal{W}_k$, a set of candidate workers $\mathcal{C}_k$, where $\mathcal{W}_k = \emptyset$ and $\mathcal{C}_k = \mathcal{N}_k$ initially. The winner set is selected in $n_k = \lfloor B/p_k \rfloor$ iterations. In each iteration, the DBA finds worker j from $\mathcal{C}_k$ with

$$j = \arg\max_{j \in \mathcal{C}_k} f(\mathcal{W}_k \bigcup \{j\}) - f(\mathcal{W}_k).$$

In other words, the measurement from winner j is expected to give the maximum K-var reduction among all candidate workers. The DBA then moves worker j from candidate set to the winner set, i.e., $\mathcal{W}_k = \mathcal{W}_k \bigcup \{j\}$ and $\mathcal{C}_k = \mathcal{C}_k \setminus \{j\}$. The algorithm terminates after n_k iterations or $\mathcal{C}_k$ is empty, whichever happens the first.

After computing the all possible winner sets $\{\mathcal{W}_k|p_k \in P\}$ using the greedy algorithm, the DBA chooses the final winner set and corresponding payment price using the exponential mechanism to guarantee differential privacy for workers' bids. As discussed in Section 3.3.3, applying the exponential mechanism requires a score function along with its global sensitivity. Here we choose the objective function $f(\cdot)$ as the score function, whose global sensitivity is the maximum change that can be caused by the change in a single bid. In particular, let us represent the greedy algorithm as a function $g(\cdot)$ that takes a bid profile b, a budget B, and a possible payment price p_k as input and outputs a winner set $\mathcal{W}_k$. The function $f \circ g$, i.e., the composition of functions f and g, then maps a bid profile (along with a budget and a payment price) into corresponding K-var reduction. Denote by Δf the *global sensitivity* of $f \circ g$, which we will derive in Section 3.4.3. Given all winner sets $\{\mathcal{W}_k|p_k \in P\}$, the DBA first calculates the probability distribution

$$\mathsf{Pr}[p = p_k] = \frac{\exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}{\sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}$$

for all $p_k \in P$, where ϵ is the privacy budget.

The DBA finally chooses the final payment price p_k and corresponding winner set $\mathcal{W}_k$ according the computed probability distribution.

We summarize the whole procedure in Algorithm 1. For each feasible payment price $p_k \in P$, the algorithm firstly initializes the winner $\mathcal{W}_k$ as an empty set and the candidate worker set as all the workers whose bids are not higher than payment price p_k (line 2). We then calculate the corresponding winner set $\mathcal{W}_k$ for the payment p_k using the greedy algorithm (line 3-6). To achieve differential privacy, we randomly output price for each winner set according to the distribution (line 8-10) and finally return the winner set and the corresponding payment price (line 11-12).

3.4.3 Global Sensitivity $\triangle f$

We now estimate Δf, the global sensitivity of function $f \circ g$. Directly estimating the global sensitivity of $f(\cdot)$ is unfortunately difficult due to the unpredictable behavior

Algorithm 1: Winner and payment price determination

input : Bid-location profile $b = (b_1, \ldots, b_n)$ and $\mathcal{X} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, budget B, and privacy budget ϵ, global sensitivity Δf

output: Winner set $\mathcal{W}$ and payment price p

foreach $p_k \in P$ **do**
  $\mathcal{W}_k \leftarrow \emptyset$, $\mathcal{C}_k \leftarrow \{j | b_j \leq p_k\}$;
  while $|\mathcal{W}_k| \leq \lfloor B/p_k \rfloor$ **do**
    Find worker j with $j = \arg\max_{j \in \mathcal{C}_k} f(\mathcal{W}_k \bigcup \{j\}) - f(\mathcal{W}_k)$;
    $\mathcal{W}_k \leftarrow \mathcal{W}_k \bigcup \{j\}$, $\mathcal{C}_k \leftarrow \mathcal{C}_k \setminus \{j\}$;
  end
end
foreach $p_k \in P$ **do**
  Calculate $\mathsf{Pr}[p = p_k] = \frac{\exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}{\sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}$;
end
Randomly pick a price p_k along with winner set $\mathcal{W}_k$ according to the computed distribution;
return $\langle \mathcal{W}_k, p_k \rangle$;

of the greedy algorithm. Instead, we seek to derive an upper bound of Δf that suffices to provide differential privacy guarantee.

Theorem 2. *Let b and b' be any two bid profiles that differ in a single bid. For any payment price $p_k \in P$, let $\mathcal{W}_k$ and $\mathcal{W}'_k$ be the winner sets chosen by the greedy algorithm based on b and b', respectively. We have*

$$|f(\mathcal{W}_k) - f(\mathcal{W}'_k)| \leq (\lfloor B/p_{\min} \rfloor / e + 1)\varphi \,, \tag{3.11}$$

where $\varphi = \max_{i \in \mathcal{N}} f(\{i\})$.

Proof. Let $\mathcal{C}_k$ and $\mathcal{C}'_k$ be the candidate sets for any payment price $p_k \in P$ under bid profiles b and b', respectively. Since b and b' differ in a single bid, $\mathcal{C}_k$ and $\mathcal{C}'_k$ differ in at most one element. Without loss of generality, suppose that $\mathcal{C}_k = \mathcal{C}'_k \bigcup \{j\}$, e.g., worker j is excluded from $\mathcal{C}'_k$ because $b_j \leq p_k < b'_j$.

We now consider the following subset selection problem.

$$\begin{aligned} \text{Maximize} \quad & f(\mathcal{W}) \\ \text{subject to} \quad & \mathcal{W} \subseteq \mathcal{C}, \\ & |\mathcal{W}| = n_k, \end{aligned} \tag{3.12}$$

where $n_k = \lfloor B/p_k \rfloor$ is the number of winners chosen by the greedy algorithm.

Let $\mathcal{W}_{\text{opt},k}$ and $\mathcal{W}'_{\text{opt},k}$ be the optimal winner sets chosen from $\mathcal{C}_k$ and $\mathcal{C}'_k$, respectively. Also let $\mathcal{W}_k$ and $\mathcal{W}'_k$ be the winner sets chosen from $\mathcal{C}_k$ and $\mathcal{C}'_k$ by the greedy algorithm, respectively. Since $\mathcal{C}'_k \subset \mathcal{C}_k$, we have $\mathcal{W}'_{\text{opt},k} \subset \mathcal{C}_k$, and therefore $f(\mathcal{W}_{\text{opt},k}) \geq f(\mathcal{W}'_{\text{opt},k})$.

Since function $f(\cdot)$ is non-negative, monotone, and submodular, the greedy algorithm can produce a solution within $(1 - \frac{1}{e})$ of the optimal solution. We therefore have

$$(1 - \frac{1}{e}) f(\mathcal{W}_{\text{opt},k}) \leq f(\mathcal{W}_k) \leq f(\mathcal{W}_{\text{opt},k}),$$

and

$$(1 - \frac{1}{e}) f(\mathcal{W}'_{\text{opt},k}) \leq f(\mathcal{W}'_k) \leq f(\mathcal{W}'_{\text{opt},k}).$$

It follows that

$$\begin{aligned} |f(\mathcal{W}_k) - f(\mathcal{W}'_k)| \leq \max(& f(\mathcal{W}'_{\text{opt},k}) - \Big(1 - \frac{1}{e}\Big) f(\mathcal{W}_{\text{opt},k}), \\ & f(\mathcal{W}_{\text{opt},k}) - \Big(1 - \frac{1}{e}\Big) f(\mathcal{W}'_{\text{opt},k})) \\ = & f(\mathcal{W}_{\text{opt},k}) - \Big(1 - \frac{1}{e}\Big) f(\mathcal{W}'_{\text{opt},k}), \end{aligned}$$

where the last equation holds because $f(\mathcal{W}'_{\text{opt},k}) \leq f(\mathcal{W}_{\text{opt},k})$.

Let $\varphi = \max_{i \in \mathcal{N}} f(\{i\})$ be the maximal K-var reduction caused by a single worker among all workers. Since $f(\cdot)$ is submodular, we have

$$\begin{aligned} f(\mathcal{W}_{\text{opt},k}) &\leq f(\mathcal{W}'_{\text{opt},k} \bigcup \{j\}) \\ &\leq f(\mathcal{W}'_{\text{opt},k}) + f(\{j\}) \leq f(\mathcal{W}'_{\text{opt},k}) + \varphi \end{aligned}$$

In addition, since $f(\mathcal{W}_{\text{opt},k}) \leq n_k\varphi$ and $n_k \leq \lfloor B/p_{\min} \rfloor$, it follows that

$$\begin{aligned}
|f(\mathcal{W}_k) - f(\mathcal{W}'_k)| &\leq f(\mathcal{W}'_{\text{opt},k}) + \varphi - \Big(1 - \frac{1}{e}\Big) f(\mathcal{W}'_{\text{opt},k}) \\
&= \frac{1}{e} f(\mathcal{W}'_{\text{opt},k}) + \varphi \leq \Big(\frac{n_k}{e} + 1\Big)\varphi \\
&\leq (\lfloor B/p_{\min} \rfloor / e + 1)\varphi.
\end{aligned}$$

□

3.5 Theoretical Analysis

We first have the following theorems regarding DPS's differential privacy guarantee.

Theorem 3. *The DPS auction mechanism is ϵ-differentially private.*

Proof. Let b and b' be two bid profiles that differ in only one worker's bid. For any payment price $p_k \in P$, let $\mathcal{W}_k$ and $\mathcal{W}'_k$ be the winner sets chosen by the greedy algorithm based on b and b', respectively.

$$\begin{aligned}
\frac{\mathsf{Pr}(M(b) = p_k)}{\mathsf{Pr}(M(b') = p_k)} &= \frac{\frac{\exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}{\sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}}{\frac{\exp\left(\frac{\epsilon f(\mathcal{W}'_k)}{2\Delta f}\right)}{\sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}'_k)}{2\Delta f}\right)}} \\
&= \frac{\exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}{\exp\left(\frac{\epsilon f(\mathcal{W}'_k)}{2\Delta f}\right)} \cdot \frac{\sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}'_k)}{2\Delta f}\right)}{\sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)} \\
&= \exp\left(\frac{\epsilon(f(\mathcal{W}_k) - f(\mathcal{W}'_k))}{2\Delta f}\right) \cdot \frac{\sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}'_k)}{2\Delta f}\right)}{\sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)} \qquad (3.13) \\
&\leq \exp\left(\frac{\epsilon \Delta f}{2\Delta f}\right) \cdot \frac{\sum_{p_k \in P} \exp\left(\frac{\epsilon(f(\mathcal{W}_k) + \Delta f)}{2\Delta f}\right)}{\sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)} \\
&= \exp\left(\frac{\epsilon}{2}\right) \cdot \frac{\exp\left(\frac{\epsilon \Delta f}{2\epsilon \Delta f}\right) \sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}{\sum_{p_k \in P} \exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)} \\
&= \exp\left(\frac{\epsilon}{2}\right) \cdot \exp\left(\frac{\epsilon}{2}\right) \\
&= \exp(\epsilon)\ .
\end{aligned}$$

□

The following theorem is about DPS's budget feasibility.

Theorem 4. *The DPS auction mechanism is budget feasible.*

Proof. For any output winner set and payment $\langle \mathcal{W}_k, p_k \rangle$\$, they satisfies $|\mathcal{W}_k| \leq \lfloor B/p_k \rfloor$ according to the Algorithm. 1. Then the total payment is $|\mathcal{W}_k| \times p_k \leq B$ □

Let $\Delta p = p_{\max} - p_{\min}$. We have the following theorem regarding the truthfulness of the auction mechanism.

Theorem 5. *The DPS auction is $\epsilon\Delta p$-truthful.*

Proof. Consider an arbitrary worker $j \in \mathcal{N}$ whose true valuation of the sensing cost is u_j. Let b and b' be two bid profiles that differ in only worker j's bid, e.g., j bids u_j and $b_j \neq u_j$ in b and b', respectively. Similar to the proof of Theorem 3, for any $p_k \in P$, we have $\mathsf{Pr}(M(b) = p_k) \geq \exp(-\epsilon)\mathsf{Pr}(M(b') = p_k)$.

It follows that

$$\begin{aligned}
\mathbb{E}_{p_k \sim M(b)}[u_i(p_k)] &= \sum_{p_k \in P} u_i(p)\mathsf{Pr}(M(b) = p_k) \\
&\geq \sum_{p_k \in P} u_i(p_k)\mathsf{Pr}(M(b') = p_k) \\
&= \exp(-\epsilon)\mathbb{E}_{p_k \sim M(b')}[u_i(p_k)] \\
&\geq (1-\epsilon)\mathbb{E}_{p_k \sim M(b')}[u_i(p_k)] \\
&= \mathbb{E}_{p_k \sim M(b')}[u_i(p_k)] - \epsilon\mathbb{E}_{p_k \sim M(b')}[u_i(p_k)].
\end{aligned} \tag{3.14}$$

Since $u_i(p_k) \leq p_{\max} - p_{\min} = \Delta p$. We have

$$\mathbb{E}_{p_k \sim M(b)}[u_i(p_k)] \geq \mathbb{E}_{p_k \sim M(b')}[u_i(p_k)] - \epsilon\Delta p.$$

We therefore conclude that our DPS auction is $\epsilon\Delta p$-truthful. □

The next theorem is about DPS's computational efficiency.

Theorem 6. *Under DPS, the winner set and payment price can be computed in polynomial time.*

Proof. We first measure the computational complexity of DPS in terms of the number of calls to function $f(\cdot)$. For each payment price $p_k \in P$, we need choose at most N winners, one in each iteration. In each iteration, finding the worker with the maximum K-var reduction takes $O(N)$ time. The complexity of computing all possible winner sets is thus $O(|P|N^2)$. In addition, the evaluation of $f(\cdot)$ takes polynomial time in terms of the total number of spectrum measurements. Therefore, the winner set and payment price can be computed in polynomial time. □

DPS is also individually rational.

Theorem 7. *DPS achieves individually rationality.*

Proof. Under DPS, each worker i's utility is $p_k - v_i$ if he is selected as a winner and paid p_k and zero otherwise. For any final payment price p_k, only the workers whose bid is lower than p_k can be selected. A worker cannot receive negative utility if he bids truthfully. Therefore, DPS is individually rational. □

Finally, we have the following theorem regarding the quality of the REM produced by the auction mechanism.

Theorem 8. *Let $\mathcal{W}_{opt}$ be the optimal winner set among all possible winner sets $\{\mathcal{W}_p|p \in P\}$. Assume that Algorithm 1 outputs a winner set $\mathcal{W}_k$ with payment price p_k. The expected average K-var reduction given by $\mathcal{W}_k$ and the maximum average K-var reduction given by $f(\mathcal{W}_{opt})$ satisfies that*

$$\mathbb{E}_{p_k \in P}[f(\mathcal{W}_k)] \geq f(\mathcal{W}_{opt}) - \ln\left(e + \frac{\epsilon|P|f(\mathcal{W}_{opt})}{2\Delta f}\right) \times \left(\frac{6\Delta f}{\epsilon}\right).$$

Proof. We start by defining the following four sets for any constant $t > 0$, including $\mathcal{B}_t = \{p_k|f(\mathcal{W}_k) > f(\mathcal{W}_{\text{opt}}) - t\}, \bar{\mathcal{B}}_t = \{p_k|f(\mathcal{W}_k) \leq f(\mathcal{W}_{\text{opt}}) - t\}, \mathcal{B}_{2t} = \{p_k|f(\mathcal{W}_k) > f(\mathcal{W}_{\text{opt}}) - 2t\}$, and $\bar{\mathcal{B}}_{2t} = \{p_k|f(\mathcal{W}_k) \leq f(\mathcal{W}_{\text{opt}}) - 2t\}$.

Since $\mathsf{Pr}[p_k \in \mathcal{B}_t] \leq 1$, we have

$$\begin{aligned}
\mathsf{Pr}[p_k \in \bar{\mathcal{B}}_{2t}] &\leq \frac{\mathsf{Pr}[p_k \in \bar{\mathcal{B}}_{2t}]}{\mathsf{Pr}[p_k \in \mathcal{B}_t]} \\
&= \frac{\sum_{p_k \in \bar{\mathcal{B}}_{2t}} \frac{\exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}{\sum_{p_i \in P} \exp\left(\frac{\epsilon f(\mathcal{W}_i)}{2\Delta f}\right)}}{\sum_{p_k \in \mathcal{B}_t} \frac{\exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}{\sum_{p_i \in P} \exp\left(\frac{\epsilon f(\mathcal{W}_i)}{2\Delta f}\right)}} \\
&= \frac{\sum_{p_k \in \bar{\mathcal{B}}_{2t}} \exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)}{\sum_{p_k \in \mathcal{B}_t} \exp\left(\frac{\epsilon f(\mathcal{W}_k)}{2\Delta f}\right)} \\
&< \frac{|\bar{\mathcal{B}}_{2t}| \exp\left(\frac{\epsilon (f(\mathcal{W}_{\text{opt}}) - 2t)}{2\Delta f}\right)}{|\mathcal{B}_t| \exp\left(\frac{\epsilon (f(\mathcal{W}_{\text{opt}}) - t)}{2\triangle f}\right)} \\
&= \frac{|\bar{\mathcal{B}}_{2t}|}{|\mathcal{B}_t|} \exp\left(\frac{-\epsilon t}{2\Delta f}\right).
\end{aligned}$$

Since $\mathsf{Pr}[p_k \in \bar{\mathcal{B}}_{2t}] + \mathsf{Pr}[p_k \in \mathcal{B}_{2t}] = 1$, it follows that

$$\begin{aligned}
\mathsf{Pr}[p_k \in \mathcal{B}_{2t}] &= 1 - \mathsf{Pr}[p_k \in \mathcal{B}_{2t}] \\
&> 1 - \frac{|\bar{\mathcal{B}}_{2t}|}{|\mathcal{B}_t|} \exp\left(\frac{-\epsilon t}{2\Delta f}\right).
\end{aligned}$$

We can estimate the $\mathbb{E}_{p_k \in P}[f(\mathcal{W}_k)]$ as

$$\begin{aligned}
\mathbb{E}_{p_k \in P}[f(\mathcal{W}_k)] &= \sum_{p_k \in P} f(\mathcal{W}_k) \mathsf{Pr}[p = p_k] \\
&\geq \sum_{p_k \in \mathcal{B}_{2t}} f(\mathcal{W}_k) \mathsf{Pr}[p = p_k] \\
&\geq (f(\mathcal{W}_{\text{opt}}) - 2t) \mathsf{Pr}[p_k \in \mathcal{B}_{2t}] \\
&\geq (f(\mathcal{W}_{\text{opt}}) - 2t) \left(1 - \frac{|\bar{\mathcal{B}}_{2t}|}{|\mathcal{B}_t|} \exp\left(\frac{-\epsilon t}{2\Delta f}\right)\right) \\
&= (f(\mathcal{W}_{\text{opt}}) - 2t) \left(1 - |P| \exp\left(\frac{-\epsilon t}{2\Delta f}\right)\right),
\end{aligned} \tag{3.15}$$

where the last inequality holds as $|\mathcal{B}_{2t}| \leq |P|$ and $|\mathcal{B}_t| \geq 1$.

For any t that satisfies the following inequality

$$t \geq \ln\left(\frac{f(\mathcal{W}_{\text{opt}})|P|}{t}\right) \times \left(\frac{2\Delta f}{\epsilon}\right), \tag{3.16}$$

we have

$$\begin{aligned}\exp\left(\frac{-\epsilon t}{2\Delta f}\right) &\leq \exp\left(\frac{-\epsilon\left(\ln\left(\frac{f(\mathcal{W}_{\text{opt}})|P|}{t}\right)\times\left(\frac{2\Delta f}{\epsilon}\right)\right)}{2\Delta f}\right)\\ &= \frac{t}{f(\mathcal{W}_{\text{opt}})|P|}.\end{aligned} \tag{3.17}$$

Plugging Inequality. (3.17) into Eq. (3.15), we get

$$\begin{aligned}\mathbb{E}_{p_k\in P}[f(\mathcal{W}_k)] &= (f(\mathcal{W}_{\text{opt}})-2t)\left(1-|P|\exp\left(\frac{-\epsilon t}{2\Delta f}\right)\right)\\ &\geq (f(\mathcal{W}_{\text{opt}})-2t)\left(1-|P|\cdot\frac{t}{f(\mathcal{W}_{\text{opt}})|P|}\right)\\ &= f(\mathcal{W}_{\text{opt}})-3t+\frac{2t^2}{f(\mathcal{W}_{\text{opt}})}\\ &> f(\mathcal{W}_{\text{opt}})-3t,\end{aligned} \tag{3.18}$$

if Inequality (3.16) holds.

We now show that $t=\ln\left(e+\frac{\epsilon|P|f(\mathcal{W}_{\text{opt}})}{2\Delta\phi}\right)\times\left(\frac{2\Delta\phi}{\epsilon}\right)$ satisfies Inequality (3.16). In particular, since $\ln\left(e+\frac{\epsilon|P|f(\mathcal{W}_{\text{opt}})}{2\Delta\phi}\right)>1$, we have $t>\frac{2\Delta\phi}{\epsilon}$. In addition, since $\ln\left(e+\frac{\epsilon|P|f(\mathcal{W}_{\text{opt}})}{2\Delta\phi}\right)>\ln\left(\frac{\epsilon|P|f(\mathcal{W}_{\text{opt}})}{2\Delta\phi}\right)$, we have

$$\begin{aligned}t &= \ln\left(e+\frac{\epsilon|P|f(\mathcal{W}_{\text{opt}})}{2\Delta f}\right)\times\left(\frac{2\Delta f}{\epsilon}\right)\\ &\geq \ln\left(|P|f(\mathcal{W}_{\text{opt}})\cdot\frac{\epsilon}{2\Delta f}\right)\times\left(\frac{2\Delta f}{\epsilon}\right)\\ &> \ln\left(\frac{|P|f(\mathcal{W}_{\text{opt}})}{t}\right)\times\left(\frac{2\Delta f}{\epsilon}\right).\end{aligned}$$

Finally, substituting $t=\ln\left(e+\frac{\epsilon|P|f(\mathcal{W}_{\text{opt}})}{2\Delta\phi}\right)\times\left(\frac{2\Delta\phi}{\epsilon}\right)$ into Eq. (3.18), we obtain

$$\mathbb{E}_{p_k\in P}[f(\mathcal{W}_k)]\geq f(\mathcal{W}_{\text{opt}})-\ln\left(e+\frac{\epsilon|P|f(\mathcal{W}_{\text{opt}})}{2\Delta f}\right)\times\left(\frac{6\Delta f}{\epsilon}\right).$$

The theorem is therefore proved. □

3.6 Simulation Results

In this section, we evaluate the performance of our auction mechanism via simulation using a real spectrum measurement dataset.

Table 3.1: Default Simulation Settings

Para.	Val.	Description.
$p_{\min}$	1	The lowest payment price
$p_{\max}$	2	The highest payment price
$b_{\min}$	2	The lowest bid price
$b_{\max}$	2	The highest bid price
$\|P\|$	101	The number of possible payment prices
$\|\mathcal{S}\|$	5	The number of dedicated sensors
ϵ	0.1	Privacy budget
$\|\mathcal{N}\|$	140	The number of crowdsourcing workers
B	30	Budget

3.6.1 Dataset

As in [29, 80], we use the CRAWDAD `cu/wimax` dataset [59] for our simulation studies. The `cu/wimax` dataset was collected at the University of Colorado Boulder (UC) and contains the signal-to-interference-plus-noise ratio (CINR) measurements of five WiMax base stations serving the University of Colorado campus. The measurements were taken by a portable spectrum analysis on a 100m equilateral triangular lattice. For our purpose, we chose the total 145 measurements for channel 308 and BSID 3674210305.

3.6.2 Simulation Settings

We randomly divide the total 145 measurements into a set of 5 measurements as the ones reported by dedicated anchor sensors and a set of the remaining 140 as submitted by mobile crowdsourcing workers. We fit the semivariogram from the total 145 measurements along with the locations where they are taken. We also assume that the semivariogram of each location in OK is inherent constant and it has been known to the DBA. In addition, for each mobile worker, their bid price is randomly picked among $(1, 1.01, \cdots, 2)$. Every point in the subsequent figures is the average of 100 runs, each with a distinct seed. Table 3.1 summarizes our default simulation settings unless mentioned otherwise.

Since DPS is the first proposal for crowdsourcing-based REM construction, we compare the performance of DPS with other two strategies.

- **Baseline differentially private auction (BDPA)**: In BDPA, for each possible price $p_k \in P$, the DBA first computes predicted average K-var reduction $f(\{i\})$ for each worker $i \in \mathcal{N}_k$ and selects winner set $\mathcal{W}_k$ as the $\lfloor B/p_k \rfloor$ workers with the highest average K-var reductions. The final winner set and payment price are then chosen using the exponential mechanism as in Algorithm 1 line 11. It is easy to verify that BDPA achieves approximate truthfulness and ϵ differential privacy as DPS.
- **Optimal single-price auction (OSPA)**: In OSPA, for each possible price $p_k \in P$, the DBA chooses the corresponding winner set $\mathcal{W}_k$ using the greedy algorithm as in Algorithm 1 and then selects the final winner set with corresponding payment price as the one that gives the maximum average K-var reduction. The K-var reduction achieved by OSPA can be viewed as the upper bound of the DPS.

We use three metrics to evaluate the performance of DPS: *average K-var reduction*, *individual worker's utility*, and *privacy leakage*. Besides the average K-var reduction and individual worker's utility defined in Section II, the privacy leakage is defined as follows.

Privacy Leakage. We use the Kullback-Leibler divergence [84] to evaluate the privacy leakage of DPS. Let b and b' be two bid profiles that differ in a single bid. Denote their payment probability distributions under DPS as $\mathsf{Pr}(M(b))$ and $\mathsf{Pr}(M(b'))$, respectively. The privacy leakage in terms of the Kullback-Leibler divergence of the two probability distribution is defined as

$$\begin{aligned}\mathsf{PL} &= KL\big(\mathsf{Pr}(M(b))|\mathsf{Pr}(M(b'))\big) \\ &= \sum_{p_k \in P} \mathsf{Pr}(M(b) = p_k) \ln\left(\frac{\mathsf{Pr}(M(b) = p_k)}{\mathsf{Pr}(M(b') = p_k)}\right).\end{aligned}$$

KL divergence indicates the statistical difference between two probability distributions. Generally, the smaller value of KL, the harder to distinguish the two bid profiles and

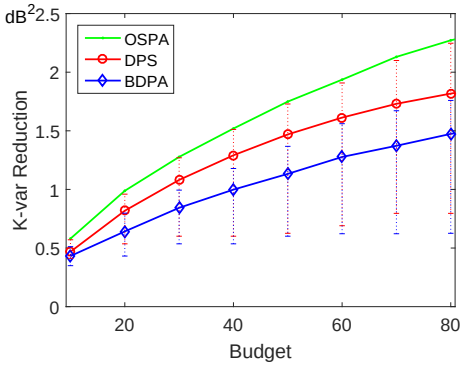


Figure 3.1: K-var reduction vs. budget B.

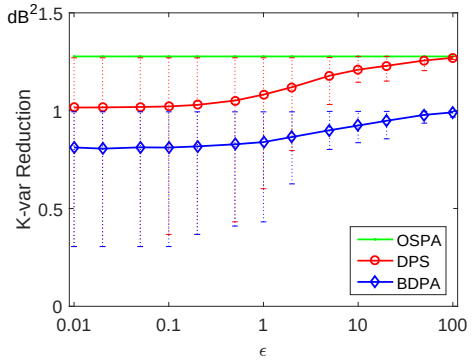


Figure 3.2: K-var reduction vs. privacy budget ϵ.

thus better protection of workers' bid privacy. Thus, PL can quantify the privacy preserving performance.

3.6.3 Simulation Results

We now report our simulation results.

3.6.3.1 Impact of Budget B

Fig. 3.1 compares the K-var reductions under BDPA, OSPA, and DPS with total budget B varying from 10 to 80. As we can see, as the total budget increases, the average K-var reductions of all three mechanisms increase. This is anticipated, as the higher budget, the more winners chosen by the DBA, the higher average K-var reduction, and vice versa. Moreover, the OSPA's K-var reduction is always the highest, which confirms that it is the upper bound of the DPS mechanism. While DPS's average K-var reduction is slightly lower than that of OSPA, it outperforms BDPA by a large margin. These results indicate that DPS can achieve approximate maximal K-var reduction while providing differential bid privacy to crowdsourcing workers.

3.6.3.2 Impact of Privacy Budget ϵ

Fig. 3.2 compares the K-var reductions of BDPA and DPS varying with privacy budget ϵ, where the K-var reduction of AMNDP is not affected by the change in ϵ and

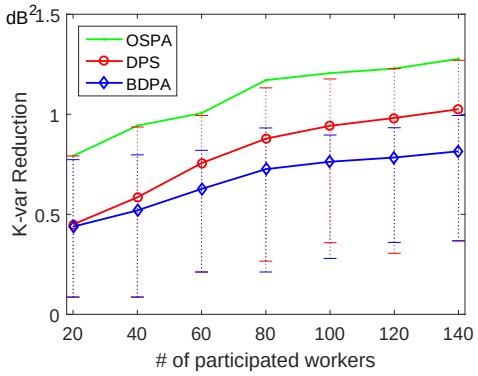
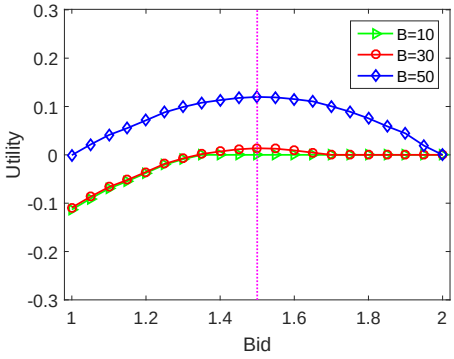


Figure 3.3: K-var reduction vs. # of workers.

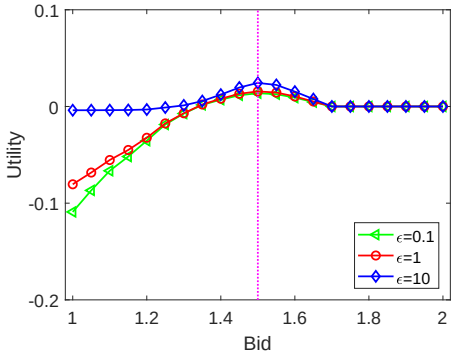
is plotted for reference only. As we can see, the K-var reductions of DPS and BDPA both increase as ϵ increases. The reason is that the larger ϵ, the higher the probability of a high-quality winner set and payment price being selected by the exponential mechanism, the higher K-var reduction, and vice versa. Moreover, the variance of K-var reductions of both BDPA and DPS decrease as ϵ increases, which is anticipated. In addition, the K-var of DPS is always higher than that of BDPA by a large margin, which confirms the effectiveness of the greedy algorithm in selecting high-quality winner set.

3.6.3.3 Impact of the Number of Workers

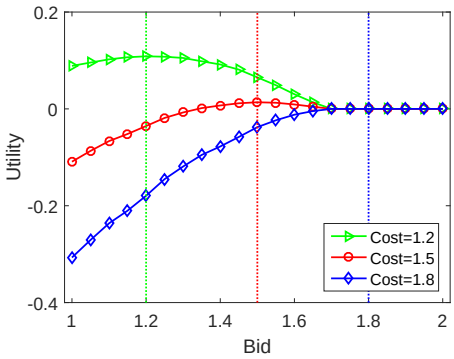
Fig. 3.3 compares the K-var reductions of BDPA, OSPA, and DPS as the number of participating workers increases from 20 to 140. We can see that the K-var reductions of all three mechanisms increase as the number of participating workers increases, as the DBA can select more winners. Similar to Fig. 3.1, the OSPA's K-var reduction is always the highest, followed by DPS, and that of BDPA is the lowest. It is worth noting that when the number of participating workers is small, the advantage of DPS over OSPA is small. For example, when the number of participating workers is 20, DPS and OSPA have the same K-var reduction. This is because the DBA can afford to select all the workers as winners in such cases. Finally, the difference between DPS and AMNDP is caused by the exponential mechanism and can be viewed as the cost of providing differential bid privacy.



(a) Utility vs. bid with different budget



(b) Utility vs. bid with different ϵ



(c) Utility vs. bid with different true cost

Figure 3.4: Expected utility of individual worker under different bid prices

3.6.3.4 Truthfulness

While we have proved that DPS is $\epsilon\Delta p$-truthful in Theorem 5, $\epsilon\Delta p$ is an upper bound of the utility that any individual worker can achieve by not bidding truthfully. In this set of experiments, we randomly select one worker, fix his true valuation of the task, and vary his bidding price. For each bid price, we calculate the probability distribution of the final winner set and payment price to obtain the worker's expected utility, i.e., $\sum_{p_k \in P} u(p_k)$.

Fig. 3.4 illustrates the individual worker's expected utility with his bidding price varying from 1 to 2. Specifically, Figs. 3.4a to 3.4c show the individual worker's expected utility under different budgets, privacy budget ϵs, and true valuations of the sensing cost, respectively. In all three figures, the vertical dotted lines represent the true valuations of the sensing cost. In all three figures, we can see that the maximum

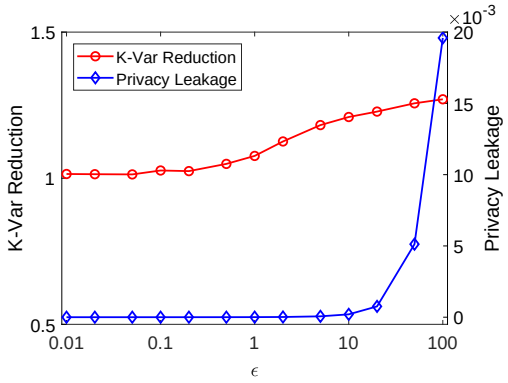


Figure 3.5: K-var reduction and privacy leakage vs. ϵ.

expected utility of the worker is achieved when he bids truthfully. While these results cannot prove that DPS is fully truthful, they nevertheless indicate that individual worker is very unlikely to increase his utility by deviating from his true valuation.

3.6.3.5 Privacy Leakage

Fig. 3.5 shows the privacy leakage and K-var reduction under DPS varying with privacy budget ϵ. As we can see, as ϵ increases, the privacy leakage and K-var reduction both increase. This is expected, as the larger ϵ, the higher the probability of high-quality winner set being selected by the exponential mechanism, the higher K-var reduction, and vice versa. At the same time, the higher ϵ, the less privacy protection, and the larger privacy leakage, and vice versa. Generally speaking, the choice of ϵ represents a trade-off between the quality of the winner set (i.e., REM's accuracy) and privacy leakage.

3.7 Summary

In this chapter, we have introduced the design and evaluation of DPS, a novel differentially-private reverse auction mechanism for crowdsourced REM construction. We have proved that the proposed auction mechanism achieves approximate truthfulness, differential privacy, and near-optimal REM accuracy. Extensive simulation studies using a real spectrum measurement dataset confirm the efficacy and efficiency of the proposed mechanism.

Chapter 4

SECURE EDGE COMPUTING-BASED SPECTRUM ACCESS REQUEST PROCESSING

4.1 Introduction

In a database-driven DSS system, the DBA potentially needs to process a large number of spectrum access requests from many SUs in real-time. A naive approach is to relying on a single DBA server to process all the requests in centralized fashion, which would not only place a high burden on the DBA's processing capability, but also may result in higher processing latency due to the large distance between some SUs and the DBA. Moreover, such an approach also suffers from a single point of failure and reduces the reliability of the DSS system. A more promising approach is to embrace the emerging edge computing paradigm by outsourcing the processing of spectrum access requests to third-party edge computing service providers. As shown in Fig. 4.1, the DBA can proactively push spectrum availability updates to distributed edge servers at a sufficiently high frequency, and edge servers process spectrum-access requests from nearby SUs on the DBA's behalf. Exploring edge computing for spectrum access request processing cannot only greatly reduce the processing latency, but also offers much better scalability and reliability than the centralized approach.

Edge computing-based spectrum access requests processing, unfortunately, is vulnerable to untrusted edge servers. In particular, third-party edge servers cannot be fully trusted to process spectrum-access request based on the most recent spectrum availability updates for various reasons. For example, a compromised edge server may blindly grant spectrum access requests to cause harmful interference to PUs so as to damage the DBA's reputation. As another example, some edge computing service providers may discriminate some SUs, e.g., those with no long term membership

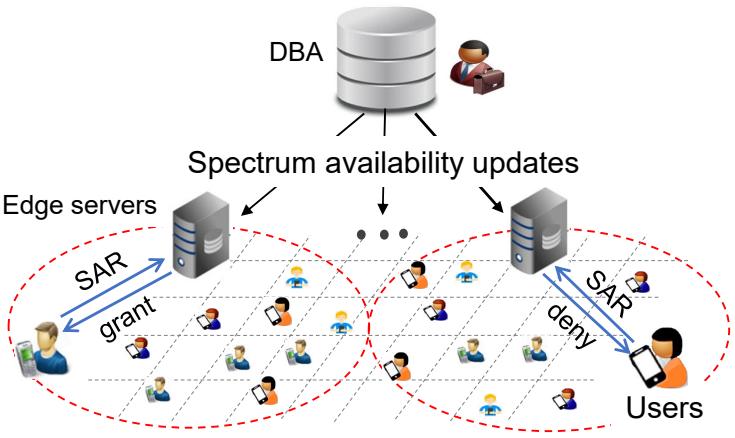


Figure 4.1: Framework of outsourced spectrum access request processing.

subscription, by denying their spectrum access requests. Last but not least, an intermittent network outage may result in an honest edge server processing spectrum access requests based on authentic but stale spectrum availability information. These situations call for sound authentication techniques to ensure any decision on spectrum access requests is based on authentic and up-to-date spectrum availability information from the DBA.

We observe that similar problems have been studied in the context of authenticated query processing in the data outsourcing paradigm, in which a data owner (e.g., the DBA) outsources its dataset and query processing to a third party service provider (e.g., the edge servers), which in turn answers data queries from users (e.g., SUs) on the data owner's behalf. While ensuring query-result authenticity, i.e., the query result contains only authentic and complete data, has been extensively studied in the past [85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96], authenticating query-result freshness has thus far received very limited attention. Common to existing solutions [97, 98, 99, 100, 101] is to divide the time into intervals and let the data owner to generate a cryptographic proof for every interval with no update. On receiving a query from a user, the third-party service provider is required to return the most recent update along with the proof of no update for every subsequent interval. Since the

size of the proof is proportional to the number of intervals after the most recent update, it is inversely proportional to the length of the interval. The state-of-art solution [101] suggests minute-based intervals to strike a good balance between communication cost and real-time guarantee, which is unable to meet the stringent real-time requirement of spectrum access requests processing. When applying these solutions to edge-computing-powered spectrum-access request processing, they suffer from either extremely high communication cost because of small time interval or inadequate real-time guarantee caused by large time interval. There is thus a pressing need to develop communication-efficient freshness authentication techniques without such limitations.

In this chapter, we tackle this challenge by introducing KV-Fresh, a novel freshness authentication techniques for edge computing-based spectrum access requests processing that can support much stronger real-time guarantee, e.g., second or millisecond-based interval, with low communication cost. Specifically, we model the spectrum updates from a DBA as a multi-version key-value store and map the problem of authenticating spectrum access request processing against untrusted edge servers into authenticated outsourced query processing. A key-value store is a database storing a collection of data records, each of which is a key-value pair that can be efficiently retrieved using the key. In a multi-version key-value store, the data value of a record (i.e., the spectrum availability at a cell) has multiple versions, each of which is an updated value received at a different time. Commercial examples of key-value stores include MongoDB, Amazon DynamoDB, Azure Cosmos DB, and so on. We consider a more general problem in which a user may ask for the most recent data record as of any given time including now. We observe that the key to meet both requirements is to break the linear dependence between the proof size and the number of intervals after the latest update. Based on this observation, we propose a novel data structure that embeds a chaining relationship among updates in different intervals to realize efficient freshness proof. Built upon this novel data structure, KV-Fresh allows the third party service provider to prove the freshness of any query result by returning information for only a small number of intervals while skipping potentially many intervals in between.

Our contributions in this chapter can be summarized as follows.

- We advocate for edge computing-based spectrum access request processing for reducing processing latency and improving the DSS system's scalability and reliability and identify the need for authenticated spectrum access request processing.
- We map the problem of authenticating spectrum access request processing against untrusted edge servers into authenticated outsourced query processing over multi-version key-value store.
- We propose a novel data structure that allows highly efficient proof of no update over a large number of intervals.
- We introduce KV-Fresh, a novel freshness authentication mechanism for outsourced spectrum availability updates that can provide stronger real-time guarantee with low communication cost.
- We confirm the high efficiency of KV-Fresh via extensive simulation studies using a synthetic dataset generated from a real dataset. In particular, our simulation results show that KV-Fresh reduces the communication cost by up to 99.6% for proving data freshness and achieves up to nine times higher throughput in comparison with the state-of-art solution INCBM-TREE [101].

The rest of this chapter is structured as follows. Section 4.2 discusses the related work. Section 4.3 formulates the problem. Section 4.4 introduces a novel data structure, LKS-MHT, and proposes an efficient freshness authentication mechanism, KV-Fresh built upon LKS-MHT. We evaluate the performance of KV-Fresh in Section 4.5 and finally conclude this chapter in Section 4.6.

4.2 Related Work

Existing solutions for authenticating data freshness in data outsourcing can be generally classified into two categories. The first category relies on the data owner to construct and maintain a proper data digest at the third party, such as a Merkle Hash tree or its variants [97, 102, 98, 103, 99]. These approaches require the data owner

to maintain large local states about historical data or incur significant cost between the data owner and third party service provider. The second category [104, 105, 106] detects the third party's misbehavior via an offline auditing process, which cannot guarantee data freshness in real time. To authenticate data freshness in real time, Yang *et al.* introduced a design based on trusted computing hardware [100]. In [101], Tang *et al.* introduced INCBM-TREE, a data structure based on the Bloom filter and multi-level key-ordered Merkle hash tree. But INCBM-TREE can only support relaxed real-time freshness check at the granularity of minute-based intervals, as the size of the freshness proof is inversely proportional to the interval length. Our proposed research is mostly related to [101] and enables freshness verification at much smaller time granularity without using trusted computing hardware.

Our work is also related to authenticating outsourced data processing [85], in which a data owner outsources its data to a third-party service provider who is responsible for answering the data queries from end users on the owner's behalf. Significant effort has been devoted to ensuring query integrity, i.e., that a query result is indeed generated from the outsourced data and contains all the data satisfying the query. Various types of queries have been studied, including relational queries [86, 87, 88], range queries [89, 90], skyline queries [91, 92, 93], top-k queries [94, 95, 96], kNN queries [107, 108], shortest-path queries [109], etc. Common to these proposals is to let the data owner outsource both its dataset and signatures to the service provider which returns both the query result and a verification object computed from the data owner's signatures, whereby the querying user can verify query-result integrity. None of these works consider the freshness of returned data records, and they are thus inapplicable to the problem addressed in this chapter.

4.3 Problem Formulation

In this section, we formulate the problem by detailing the data model, system models, adversary model, and design goals.

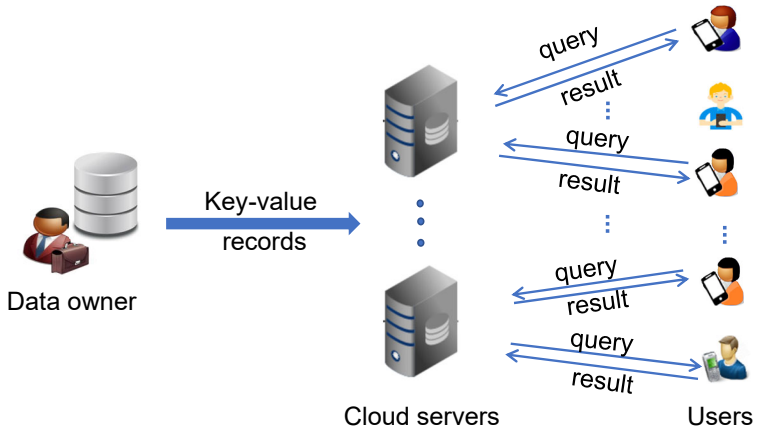


Figure 4.2: Illustration of a data outsourcing system.

4.3.1 Data Model

We model the spectrum availability information pushed by the DBA to local edge servers as a multi-version key-value store. In particular, we assume the DBA divides its service region into K non-overlapping *cells* of equal size, denoted by $\mathcal{K} = \{1, \ldots, K\}$. We assume that all the locations within the same cell have the same spectrum availability at any given time. The spectrum availability in each cell $k \in \mathcal{K}$ is represented as $V_k = \langle k, v, t \rangle$, where k is the cell ID and serves as the key, v is the latest spectrum availability in cell k, and t is the timestamp at which the latest spectrum update is received. The spectrum availability value v is updated over time and can be of different forms. For example, $v = 0$ can indicate the channel is vacant while $v = 1$ indicates the channel is busy. As another example, v can represent the PU's received signal strength based on the REM maintained at the DBA. Without loss of generality, we assume $v \in \mathbb{R}$.

4.3.2 System Model

We map the problem of authenticating outsourced spectrum access requests processing into authenticated query processing in data outsourcing. We consider a data outsourcing system consisting of three parties: a data owner (i.e., the DBA), a cloud server owned by a third-party service provider (i.e., the edge server), and many

end users (i.e., SUs). The data owner outsources a dataset in the form of a multi-version key-value store to the cloud server, which in turn answers data queries from users on behalf of the data owner.

As shown in Fig. 4.2, the data owner proactively pushes key-value records $\{\langle k, v, t\rangle | k \in \mathcal{K}\}$ to the cloud server as they become available. We assume that the data owner has limited storage and only has the most recent key-value records for any $k \in \mathcal{K}$. The cloud server maintains the collection of key-value records from the data owner, commonly referred to as multi-version key-value store, and based on which process data queries from end users on the data owner's behalf. Users access data records in the key-value store through the cloud server's GET API that supports both point queries and range queries. Specifically, a point query is represented as $Q(k, t_q)$, where k is the queried key and t_q is an optional parameter indicating the point of time up to which the data record is requested. On receiving a query $Q(k, t_q)$, the cloud server needs to return the most recent data record for key k as of t_q. Moreover, a range query is modeled by $Q([l, r], t_q)$, where $1 \leq l < r \leq |\mathcal{K}|$ and $[l, r]$ denotes the range of keys being queried. On receiving query $Q([l, r], t_q)$, the cloud server needs to return the most recent data records for every key $k \in [l, r]$ as of t_q. It is easy to see that point query is a special case of range query where $l = r$. For both point queries and range queries, the absence of the optional parameter t_q indicates that the user is asking for the most recent data record for a specific key or the most up-to-date records for a set of keys belonging to the key range as of now.

4.3.3 Adversary Model

We assume that the data owner is trusted to faithfully perform all system operations. In contrast, the cloud server cannot be fully trusted and may launch the following two attacks. First, the cloud server may return forged or tampered data records that do not belong to the data owner's dataset. Second, the cloud server may return authentic but stale data records in response to the user's GET query.

We assume that the communication channels between the data owner and the cloud server as well as between the cloud server and users are secured using standard techniques, e.g., TLS [52]. In addition, we also assume that the data owner cannot predict the keys that the user will query in advance.

4.3.4 Design Goals

Strict freshness verification—also referred to as real-time freshness check in [101]—requires the cloud server to not only push authenticated data updates to the cloud sever as soon as there are available but also constantly inform the cloud server even if there is no update, which would result in prohibitive processing and communication cost. As in the state-of-art solution in [101], we seek to achieve relaxed real-time freshness verification. Specifically, we assume that time is divided into intervals of equal length, which means that the data owner pushes authenticated data updates to the cloud server on the interval basis. To ease the presentation, we assume that in every interval, every data object $k \in \mathcal{K}$ has either no or just one new updated value. Note that our proposed mechanism can be easily adapted to support multiple updated values in one interval.

In view of the aforementioned two attacks, we aim to design a freshness authentication mechanism to allow a user to verify whether the query result returned by the cloud server satisfies the following two conditions.

- *Query-result integrity*: for each queried key k, the returned data value v is indeed an updated value from the data owner and has not been tampered with.
- *Query-result freshness*: for each queried key k, there is no update in any interval that starts after t and ends before or exactly at t_q.

In other words, we aim to achieve relaxed real-time freshness verification because it cannot guarantee no update for key k in the interval that encloses t_q. The smaller the interval size, the stronger the real-time guarantee, and vice versa. We aim to support strong real-time guarantee with millisecond-based intervals and low communication and

computation costs. In particular, the mechanism should incur low update cost between the data owner and the cloud server as well as low communication and computation cost for proving data freshness.

4.4 KV-Fresh

In this section, we first introduce two strawman approaches for freshness authentication followed by an overview of KV-Fresh. We then introduce a novel data structure that underpins KV-Fresh and its construction. Finally, we detail the design of KV-Fresh.

4.4.1 Two Strawman Approaches

We first introduce two strawman approaches to enable query-result integrity and freshness verification.

Strawman Approach 1. The first approach is to let the data owner maintain the most recent update for every key and build a Merkle hash tree over all data records in every interval, some of which are updated in the current interval and the rest are copied from the previous interval. The data owner pushes the Merkle hash tree to the cloud server. With the Merkle hash tree constructed for every interval, the cloud server can prove the integrity and freshness of the query result. This approach incurs low communication cost for proving data freshness but excessively high update cost between the data owner and cloud server, as the data owner has to transmit information for every key even if many have no update in the short interval. In particular, the update cost between the data owner and the cloud server is linear to the size of the key space.

Strawman Approach 2. The second approach is to let the cloud server construct a Key-Ordered Merkle Hash Tree (KOMT) for every interval over only keys with update, where the absence of a key implicitly indicates that the most recent update for this key happened in one of the previous intervals. Given a batch of key-value records, the data owner sorts the records according to their keys and builds a Merkle hash

tree over the sorted list. Doing so can minimize the communication cost between the data owner and the cloud server due to fewer leaf nodes in each KOMT. However, it still incurs high communication cost for proving data freshness if each key is updated infrequently, as the cloud server needs to prove that there is no update in possibly many intervals after the most recent update. More importantly, the number of intervals after the most recent update is inversely proportional to the size of interval, which means that strong real-time guarantee, i.e., small interval size, would incur significant communication cost for proving data freshness.

4.4.2 Overview Of KV-Fresh

KV-Fresh is designed to take the advantages of both strawman approaches by striking a good balance between the update cost between data owner and cloud service provider and the size of freshness proof. In particular, the first strawman approach achieves low communication cost for proving data freshness by copying the most recent update to the Merkle hash tree constructed for the current interval. Doing so allows the cloud server to prove data freshness using the Merkle hash tree constructed for the current interval. In contrast, the second approach achieves low update cost between the data owner and the cloud server by greatly reducing the number of leaf nodes of the Merkle hash tree constructed for every interval. We find that the key to realize efficient freshness authentication with strong real-time guarantee is to simultaneously maintaining small Merkle hash tree size while realizing efficient proof of no update in possibly many intervals after the most recent update.

Based on the above observation, we introduce *Linked Key Span Merkle Hash Tree (LKS-MHT)*, a novel data structure to achieve small Merkle hash tree size in every interval while allowing efficient proof of no update in possibly many intervals. The key idea behind LKS-MHT is to bundle adjacent keys with no update in one interval as a key block to reduce the number of leaf nodes. To enable efficient proof of no update over multiple intervals, each key block embeds the index of an earlier interval if none of the keys in the block has update after the earlier interval. This allows the cloud

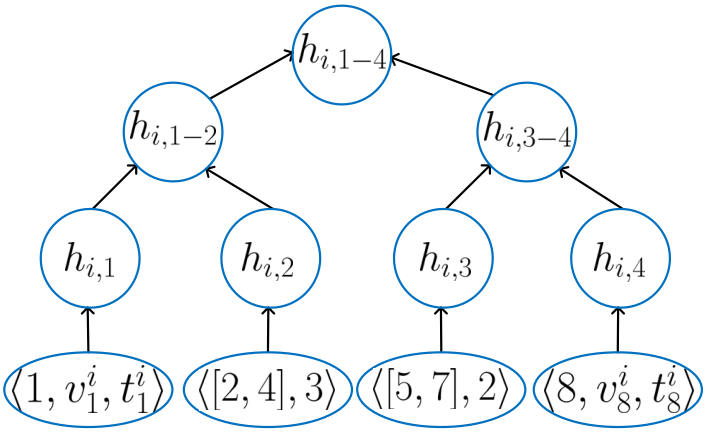


Figure 4.3: An example of LKS-MHT.

server to skip possibly many intervals in between in the freshness proof. LKS-MHT can effectively break the linear dependence between the freshness proof size and the number of intervals with no update and thus enables highly efficient freshness authentication.

Under KV-Fresh, the data owner builds one LKS-MHT for every interval and pushes the LKS-MHT to the cloud server. The LKS-MHT contains information for every key in the key space, either an updated value received in the current interval or an index of an earlier interval, for which the LKS-MHT contains the most recent update or the index of another earlier interval. On receiving a GET query from the end user, the cloud server returns the LKS-MHT leaf node containing the queried key in the queried interval. If there is no update for the key in the queried interval, the cloud server further returns the LKS-MHT leaf node for the interval with an index embedded in the leaf node of the queried interval. This process continues until the most recent update for the queried key is found. In what follows, we first introduce LKS-MHT and its construction and then detail the operations of KV-Fresh.

4.4.3 LKS-MHT:Linked Key Span Merkle Hash Tree

We now introduce LKS-MHT, the data structure that underpins KV-Fresh. An LKS-MHT T_i is a binary tree constructed for each interval i with θ_i leaf nodes $L_{i,1}, \ldots, L_{i,\theta_i}$. Every leaf node $L_{i,j}, 1 \leq j \leq \theta_i$, consists of the following fields.

(1) A key block $K_{i,j} = [l_{i,j}, r_{i,j}]$ with $l_{i,j}, r_{i,j} \in \mathcal{K}$ and $l_{i,j} \leq r_{i,j}$. If $l_{i,j} = r_{i,j}$, then $K_{i,j}$ represents a single key $l_{i,j}$.

(2.a) An interval index $\gamma_{i,j} \in \{0, \ldots, i-1\}$ that indicates that there is no update for any key in $K_{i,j}$ from interval $\gamma_{i,j}+1$ to i. In other words, the information about the most recent update for each key in $K_{i,j}$ can be found in interval $\gamma_{i,j}$ or earlier.

(2.b) Or an updated key value v_k^i along with timestamp t_k^i, if $K_{i,j}$ represents a single key k (i.e., $k = l_{i,j} = r_{i,j}$) which receives an update in interval i.

Given $L_{i,1}, \ldots, L_{i,\theta_i}$, the LKS-MHT is constructed similar to the traditional Merkle hash tree. In particular, we first calculate $h_{i,j} = H(L_{i,j})$ for all $1 \leq j \leq \theta_i$, where $H(\cdot)$ denotes a cryptographic hash function such as SHA-256. We then compute every internal node as the hash of the concatenation of its two children. Note that if the number of leaf nodes is not a perfect power of two, some dummy leaf nodes need be introduced.

Fig. 4.3 shows an example of the LKS-MHT constructed for an interval i with the key space $\mathcal{K} = \{1, \ldots, 8\}$. The first leaf node corresponds to key $K_{i,1} = 1$ with the updated value v_1^i and timestamp t_1^i received in the interval i; the second leaf node corresponds to a key block $K_{i,2} = [2, 4]$ and an interval index 3, meaning that the most recent information for keys in $[2, 4]$ can be found in interval 3 or earlier; the third leaf node corresponds a key block $K_{i,3} = [5, 7]$ and an interval index 2, meaning that the most recent information about any key in $[5, 7]$ can be found in interval 2 or earlier; and the last leaf node corresponds to key $K_{i,8} = 8$ with updated value v_8^i and timestamp t_8^i.

To see how LKS-MHT can be used to realize efficient freshness authentication, consider Fig. 4.4 as an example, where eight LKS-MHTs $T_1, \ldots, T_8$ are constructed for intervals 1 to 8 over key space $\mathcal{K} = \{1, 2, 3, 4\}$. Assume that the user issues a GET query as $\mathsf{Q}(2, t_q)$, where t_q is the end of interval 8. Since the most recent update for key 2 is v_2^3 received in interval 3, the cloud server needs to prove that there has been no update in intervals 4 to 8. To do so, the cloud server only needs to return the first leaf node in LKS-MHT T_8, which is a key block $[1, 2]$ and embeds an interval index

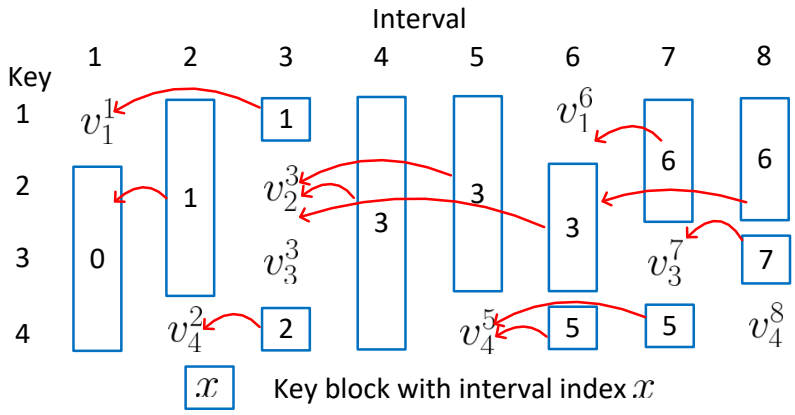


Figure 4.4: Illustration of LKS-MHT-based freshness authentication

6, and the second leaf node in LKS-MHT T_6, which is a key block $[2, 3]$ and embeds an interval index 3, and the second leaf node in LKS-MHT T_3, which is a single key 2 with updated value v_2^3. As we can see, there is no need for the cloud server to return any information about intervals 4, 5, and 7.

In the next two subsections, we introduce how to construct the LKS-MHT for the first interval and the subsequent intervals, respectively.

4.4.4 LKS-MHT Construction in the First Interval

We first show how to construct LKS-MHT T_i for the first interval i ($i = 1$). Denote by $\mathcal{K}_1 \subseteq \mathcal{K}$ the subset of keys that receive updates in the first interval. Without loss of generality, suppose $\mathcal{K}_1 = \{k_{1,1}, k_{1,2}, \ldots, k_{1,\lambda_1}\}$, where $\lambda_1 = |\mathcal{K}_1|$ and $k_{1,1} < k_{1,2} < \cdots < k_{1,\lambda_1}$. We can see that the λ_1 keys, $\mathcal{K}_1 = \{k_{1,1}, k_{1,2}, \ldots, k_{1,\lambda_1}\}$, split the whole key space $\mathcal{K} = \{1, \ldots, K\}$ into $\lambda_1 + 1$ key blocks without update, $B_1 = [1, k_{1,1} - 1]$, $B_2 = [k_{1,1} + 1, k_{1,2} - 1], \ldots, B_{\lambda_1+1} = [k_{1,\lambda_1} + 1, K]$. For simplicity, we assume that none of these key blocks are empty, from which we can form $\theta_i = 2\lambda_1 + 1$ key blocks $\{K_{1,j}\}_{j=1}^{\theta_i}$, where

$$K_{1,j} = \begin{cases} B_{(j+1)/2}, & \text{if } j \text{ is odd,} \\ k_{1,j/2}, & \text{if } j \text{ is even,} \end{cases}$$

for all $1 \leq j \leq \theta_i$. We then create one leaf node $L_{1,j}$ for each key block $K_{1,j}$, where

$$L_{1,j} = \begin{cases} \langle B_{(j+1)/2}, 0 \rangle, & \text{if } j \text{ is odd,} \\ \langle k_{1,j/2}, v^1_{k_{j/2}}, t^1_{k_{j/2}} \rangle, & \text{if } j \text{ is even.} \end{cases}$$

4.4.5 LKS-MHT Construction in Subsequent Intervals

We now discuss how to construct LKS-MHT T_i for the subsequent interval i ($i \geq 2$), for which the key question is to determine the set of key blocks with corresponding interval index. Let $\mathcal{K}_i = \{k_{i,1}, k_{i,2}, \ldots, k_{i,\lambda_i}\}$ be the subset of keys that have received updates in the subsequent interval i, where $\lambda_i = |\mathcal{K}_i|$ and $k_{i,1} < k_{i,2} < \cdots < k_{i,\lambda_i}$. For every subsequent interval i, the leaf nodes of T_i are determined jointly by the leaf nodes of T_{i-1} and $\mathcal{K}_i$ in two steps: (1) constructing a set of candidate leaf nodes and (2) determining the leaf nodes.

Candidate leaf nodes. First, we can obtain a set of candidate leaf nodes based on $L_{i-1,1}, \ldots, L_{i-1,\theta_{i-1}}$, and $\mathcal{K}_i$. Consider as an example a leaf node $L_{i-1,j}$ with key block $K_{i-1,j} = [l_{i-1,j}, r_{i-1,j}]$ and interval index $\gamma_{i-1,j} < i$. Assume that $|K_{i-1,j}| \geq 2$. If no key in $K_{i-1,j}$ receives any update in interval i, we create one candidate leaf node the same as $L_{i-1,j}$. Otherwise, we split $K_{i-1,j}$ into multiple non-overlapping key blocks and create one candidate leaf node from each of them. Each candidate leaf node either contains a key with update in interval i or a key block with no update that inherits the interval index $\gamma_{i-1,j}$ from $L_{i-1,j}$. For example, if a single key $k \in K_{i-1,j}$ is updated in interval i and $l_{i-1,j} < k < r_{i-1,j}$, we can split $K_{i-1,j}$ into three smaller candidate blocks and create three candidate leaf nodes: the first one with key block $[l_{i-1,j}, k-1]$ and the same interval index $\gamma_{i-1,j}$, the second one with a single key k and updated value v^i_k and timestamp t^i_k, and the third one with key block $[k+1, r_{i-1,j}]$ and the same interval index $\gamma_{i-1,j}$.

We summarize the general procedure for constructing a list of candidate leaf nodes in Algorithm 1, which takes a list of leaf nodes $L_{i-1,1}, \ldots, L_{i-1,\theta_{i-1}}$ of LKS-MHT T_{i-1} and $\mathcal{K}_i$ as input and outputs a sorted list of candidate leaf nodes C_i. Specifically, we initiate the list of candidate leaf nodes to an empty list (Line 1). We then create

Algorithm 2: Construct candidate leaf nodes

input : Leaf nodes $L_{i-1,1}, \ldots, L_{i-1,\theta_{i-1}}$ and $\mathcal{K}_i$
output: An ordered list of candidate leaf nodes for T_i

1 $C_i \leftarrow$ emptylist;
2 **foreach** $j \in \{1, \ldots, \theta_{i-1}\}$ **do**
3 $\quad K_{i,j} \leftarrow K_{i-1,j}$;
4 $\quad$ **if** $L_{i-1,j} = \langle k, v_k^{i-1}, t_k^{i-1} \rangle$ **then**
5 $\quad\quad \gamma_{i,j} = i - 1$;
6 $\quad$ **end**
7 $\quad$ **else if** $L_{i-1,j} = \langle [l_{i-1,j}, r_{i-1,j}], \gamma_{i-1,j} \rangle$ **then**
8 $\quad\quad \gamma_{i,j} = \gamma_{i-1,j}$;
9 $\quad$ **end**
10 $\quad$ Append $C_{i,j} = \langle K_{i,j}, \gamma_{i,j} \rangle$ to C_i;
11 **end**
12 **foreach** $k_{i,j} \in \mathcal{K}_i$ **do**
13 $\quad$ Find $C_{i,x} \in C_i$ such that $k_{i,j} \in K_{i,x}$;
14 $\quad$ Delete $C_{i,x}$ from C_i;
15 $\quad$ Insert $C_i^* = \langle k_{i,j}, v_{k_{i,j}}^i, t_{k_{i,j}}^i \rangle$ after $C_{i,x-1}$;
16 $\quad$ **if** $k_{i,j} > l_{i,x}$ **then**
17 $\quad\quad$ Insert $\langle [l_{i,x}, k_{i,j} - 1], i \rangle$ before C_i^*;
18 $\quad$ **end**
19 $\quad$ **if** $k_{i,j} < r_{i,x}$ **then**
20 $\quad\quad$ Insert $\langle [k_{i,j} + 1, r_{i,x}], i \rangle$ after C_i^*;
21 $\quad$ **end**
22 **end**
23 **return** C_i;

one candidate leaf node from each leaf node $L_{i-1,j}$ where the interval index is set to $i - 1$ if $L_{i-1,j}$ corresponds to a single key with update in interval $i - 1$ or $\gamma_{i,j-1}$ if it corresponds to a key block (Lines 2 to 11). We then check every key $k_{i,j} \in \mathcal{K}_i$ to make necessary adjustment to the candidate leaf nodes (Lines 12 to 22). Specifically, for every $k_{i,j} \in \mathcal{K}_i$, we find the candidate leaf node $C_{i,x}$ whose key block encloses $k_{i,j}$ and replace $C_{i,x}$ with a new candidate leaf node $\langle k_{i,j}, v_{k_{i,j}}^i, t_{k_{i,j}}^i \rangle$, a candidate leaf node on the left if $k_{i,j} > l_{i,x}$, and a candidate leaf node on the right if $k_{i,j} < r_{i,x}$.

Leaf nodes. We now determine the leaf nodes for T_i from the candidate leaf nodes, for which the key is to merge some adjacent candidate leaf nodes into one to maintain a small number of leaf nodes. Without merging, the number of leaf nodes would increase monotonically at every interval and eventually reach $|\mathcal{K}|$, resulting in

excessive update cost between the data owner and the cloud server as in Strawman Approach 1.

Under what condition can adjacent candidate leaf nodes be merged? We observe that multiple adjacent candidate leaf nodes can be merged into one if none of the keys in the corresponding key blocks is updated in interval i. Specifically, for a group of adjacent candidate leaf nodes $C_{i,j}, \ldots, C_{i,j+s}$ for some $s \geq 1$, if none of the keys in their respective key blocks $\bigcup_{x=j}^{j+s} K_{i,x}$ have received any update in interval i, then we can merge key blocks $K_{i,j}, \ldots, K_{i,s}$ into one and create a new leaf node as $\langle \bigcup_{x=j}^{j+s} K_{i,x}, i-1 \rangle$, which indicates that the most recent information about any key in $\bigcup_{x=j}^{j+s} K_{i,x}$ can be found in T_{i-1}.

Which adjacent candidate leaf nodes should be merged? A plausible answer is to merge every group of consecutive candidate leaf nodes into one to minimize the number of leaf nodes and thus the update cost between the data owner and the cloud server. However, doing so would increase the size of freshness proof, as the cloud server needs to return information for more intervals. Fig. 4.5 shows an example of blindly merging all possible leaf nodes for 8 LKS-MHTs. Assume that the end user issues a GET query as $Q(2, t_q)$, where t_q is at the end of interval 8. The cloud server needs to return the first leaf node of T_8, which is a key block $[1, 3]$ and embeds an interval index 7, and the first leaf node in LKS-MHT T_7, which is a key block $[1, 2]$ and embeds an interval index 6, the second leaf node of T_6, which is a key block $[2, 4]$ and embeds an interval index 5, the first leaf node in LKS-MHT T_5, which is a key block $[1, 3]$ and embeds an interval index 3, and the second leaf node of T_2 which is a single key 2 with the updated value v_2^3. In comparison with the previous example shown in Fig. 4.4, the cloud server needs to return two more leaf nodes.

We first observe that some merging decisions can be made based on whether related keys have updates in the two intervals. Let $C_i = \langle C_{i,1}, \ldots, C_{i,\phi_i} \rangle$ be the list of candidate leaf nodes output by Algorithm 1, where ϕ_i is the number of candidate leaf nodes. We define b_j as the decision variable such that $b_j = 1$ if $C_{i,j}$ and $C_{i,j+1}$ are merged into one and 0 otherwise for all $1 \leq j \leq \phi_i - 1$. We find that b_j can be

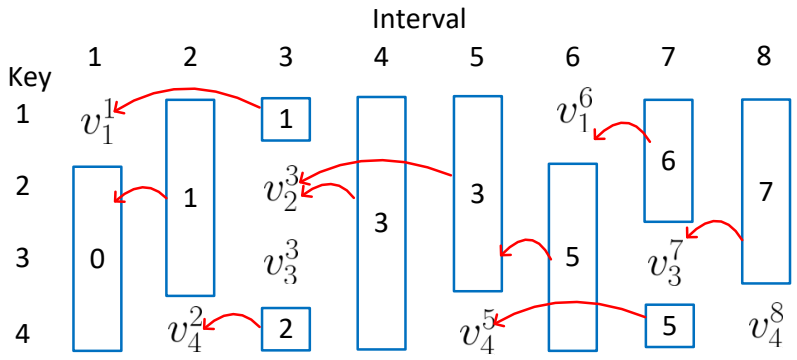


Figure 4.5: An example of LKS-MHTs constructed under maximum merging.

predetermined in the following two cases.

- **Case 1**: If either $C_{i,j}$ or $C_{i,j+1}$ corresponds to a single key that has received an update in interval i, then $b_j = 0$, as the corresponding leaf node needs to record the updated value and thus cannot be merged with the other.
- **Case 2**: If $C_{i,j}$ and $C_{i,j+1}$ each correspond to a single key that has received an update in interval $i-1$, i.e., $|K_{i,j}| = |K_{i,j+1}| = 1$ and $\gamma_{i,j} = \gamma_{i,j+1} = i-1$, then we should merge them into one, i.e., $b_j = 1$. Doing so can reduce the number of leaf nodes without increasing freshness proof size, because the cloud server needs to return the leaf node for at least one interval after the most recent update in interval $i-1$.

Based on the above observations, we define three index sets as $\Phi = \{1, \ldots, \phi_i - 1\}, \Phi_0 = \{j | j \in \Phi, K_{i,j} \in \mathcal{K}_i \vee K_{i,j+1} \in \mathcal{K}_i\}$ and $\Phi_1 = \{j | j \in \Phi, |K_{i,j}| = |K_{i,j+1}| = 1, \gamma_{i,j} = \gamma_{i,j+1} = i-1\}$, where Φ_0 and Φ_1 correspond to the first and second cases, respectively. In other words, $b_j = 0$ for all $j \in \Phi_0$ and $b_j = 1$ for all $j \in \Phi_1$. We further note that if we set $b_j = 1$ for all $j \in \Phi \setminus \Phi_0$, i.e., merging every possible pair of candidate leaf nodes, then it would take $|\Phi| - |\Phi_0|$ merging operations and the number of remaining leaf nodes is given by

$$\begin{aligned} \phi_i - (|\Phi| - |\Phi_0|) &= \phi_i - (\phi_i - 1 - |\Phi_0|) \\ &= |\Phi_0| + 1 \ . \end{aligned}$$

Therefore, the minimum number of leaf nodes that T_i can have is $|\Phi_0| + 1$.

We make the remaining merging decisions through an optimization approach. In what follows, we introduce two optimization problem formulations with different objective functions and present their solutions.

4.4.5.1 Formulation 1: Expected Freshness Proof Size Minimization

Our first formulation aims to minimize the expected size of freshness proof under the constraint of the maximum number of leaf nodes. We observe that the size of freshness proof is linear to the number of intervals for which the cloud server needs to return a leaf node in response to a point query. Let $h_{k,i}$ and $h_{k,i-1}$ denote the numbers of leaf nodes the cloud server needs to return in response to queries $Q = (k, i)$ and $Q = (k, i-1)$, respectively, for all $k \in \mathcal{K}$. Also let p_k be the probability of each key k being queried, where $\sum_{k\in\mathcal{K}} p_k = 1$. If every key is equally likely being queried, we then have $p_k = 1/\mathcal{K}$ for all $k \in \mathcal{K}$. Let $\triangle h_k = h_{k,i} - h_{k,i-1}$ for all $k \in \mathcal{K}$. The expected number of leaf nodes that the cloud server needs to return for freshness proof is given by

$$\begin{aligned} \mathbb{E}(h_i) &= \sum_{k\in\mathcal{K}} p_k h_{k,i} \\ &= \sum_{k\in\mathcal{K}} p_k h_{k,i-1} + \sum_{k\in\mathcal{K}} p_k \triangle h_k \ , \end{aligned} \tag{4.1}$$

where $\mathbb{E}(\cdot)$ denotes expectation. Since merging decisions in interval i have no impact on the first term $\sum_{k\in\mathcal{K}} p_k h_{k,i-1}$, minimizing $\mathbb{E}(h_i)$ is equivalent to minimizing $\sum_{k\in\mathcal{K}} p_k \triangle h_k$.

Next, we analyze the relationship between decision variables $b_1, \ldots, b_{\phi_i - 1}$ and $\sum_{k\in\mathcal{K}} p_k \triangle h_k$. First, we observe that $\triangle h_k = 1$ if key k belongs to a candidate leaf node being merged with another adjacent one and 0 otherwise. Let $\Phi' = \Phi \setminus (\Phi_0 \bigcup \Phi_1)$ and $\{b_j | j \in \Phi'\}$ be the remaining decision variables that need be determined. Further denote by $\Phi'_1 = \{b_j = 1 | j \in \Phi'\}$ and $\Phi'_0 = \{b_j = 0 | j \in \Phi'\}$ the subsets of decision variables set to one and zero, respectively. Given Φ'_1 and Φ_1, a candidate leaf node

$C_{i,j}$ is merged with another one if either $j-1$ or $j \in \Phi_1' \bigcup \Phi_1$. Let $\Pi = \{j | j-1 \in \Phi_1' \bigcup \Phi_1 \vee j \in \Phi_1' \bigcup \Phi_1 \wedge j \in \Phi\}$. We have

$$\sum_{k \in \mathcal{K}} p_k \triangle h_k = \sum_{j \in \Pi} \sum_{k \in K_{i,j}} p_k,$$

where $K_{i,j}$ is the key block of $C_{i,j}$.

Let $f(\Phi_1') = \sum_{j \in \Pi} \sum_{k \in K_{i,j}} p_k$. We formulate the merging decisions as the following programming problem.

$$\begin{aligned} \text{minimize} \quad & f(\Phi_1') \\ \text{subject to} \quad & \Phi_1' \subseteq \Phi', \\ & \phi_i - |\Phi_1 \bigcup \Phi_1'| \leq \max(\tau, |\Phi_0| + 1), \\ & b_j = 0, \forall j \in \Phi_0 \bigcup \Phi_0', \\ & b_j = 1, \forall j \in \Phi_1 \bigcup \Phi_1', \end{aligned} \tag{4.2}$$

where $\phi_i - |\Phi_1 \bigcup \Phi_1'|$ is the number of leaf nodes after $|\Phi_1 \bigcup \Phi_1'|$ merging operations, and τ is a system parameter that limits the number of leaf nodes for every LKS-MHT and usually set to be the larger the expected number of updates in each interval.

We now introduce an efficient greedy algorithm to solve the above optimization problem with guaranteed approximation ratio. We can see that the objective function $f : 2^{\Phi'} \to \mathbb{R}$ is a set function, and the following theorem characterizes its properties.

Theorem 9. *The objective function $f(\cdot)$ in Eq. (4.2) is non-negative, submodular, and monotone.*

Proof. First, as $p_k \geq 0$ for all $k \in \mathcal{K}$, and $\{K_{i,j} | j \in \Phi'\} \subseteq \mathcal{K}$, the objection function $f(\cdot)$ is non-negative for any $k \in \Phi'$.

Second, we show that $f : 2^{\Phi'} \to \mathbb{R}$ is submodular. Consider any two subsets Φ_x and Φ_y where $\Phi_x \subseteq \Phi_y \subset \Phi'$. There are two cases. First, if $\Phi_x = \Phi_y$, then $\Phi_x \bigcup \{j\} = \Phi_y \bigcup \{j\}$ for any $j \in \Phi'$. It follows that $f(\Phi_x \bigcup \{j\}) - f(\Phi_x) = f(\Phi_y \bigcup \{j\}) - f(\Phi_y)$.

Now let us consider the second case where $\Phi_x \subset \Phi_y$. For any $j \in \Phi' \setminus \Phi_y$, i.e., C_j and C_{j+1} are two adjacent candidate leaf nodes, there are further four possible cases.

- Case 1: if $j-1 \in \Phi_y \setminus \Phi_x$ and $j+1 \in \Phi_y \setminus \Phi_x$, then both C_j and C_{j+1} are merged with other candidate leaf nodes in Φ_y, but none of them has been merged with another in Φ_x, which indicates that $f(\Phi_y \bigcup \{j\}) - f(\Phi_y) = 0$ and $f(\Phi_x \bigcup \{j\}) - f(\Phi_x) > 0$. Thus, for any $j \in \Phi' \setminus \Phi_y$, $f(\Phi_x \bigcup \{j\}) - f(\Phi_x) > f(\Phi_y \bigcup \{j\}) - f(\Phi_y) = 0$.
- Case 2: if $j-1 \in \Phi_y \setminus \Phi_x$ and $j+1 \notin \Phi_y \setminus \Phi_x$, then C_j has been merged with another candidate leaf node C_{j-1},$j-1 \in \Phi_y$ resulting in smaller return from adding j. Thus, for any $j \in \Phi' \setminus \Phi_y$, $f(\Phi_x \bigcup \{j\}) - f(\Phi_x) > f(\Phi_y \bigcup \{j\}) - f(\Phi_y)$.
- Case 3: if $j-1 \notin \Phi_y \setminus \Phi_x$ and $j+1 \in \Phi_y \setminus \Phi_x$, then this case is symmetric to Case 2, which leads to the same conclusion that for any $j \in \Phi' \setminus \Phi_y$, $f(\Phi_x \bigcup \{j\}) - f(\Phi_x) > f(\Phi_y \bigcup \{j\}) - f(\Phi_y)$.
- Case 4: if $j-1 \notin \Phi_y \setminus \Phi_x$ and $j+1 \notin \Phi_y \setminus \Phi_x$, neither C_j nor C_{j+1} have been merged with another one in Φ_y resulting in the same return from adding j. Thus, for any $j \in \Phi' \setminus \Phi_y$ we have $f(\Phi_x \bigcup \{j\}) - f(\Phi_x) = f(\Phi_y \bigcup \{j\}) - f(\Phi_y)$.

To sum up, for any $\Phi_x \subseteq \Phi_y \subset \Phi'$ and $j \in \Phi' \setminus \Phi_y$, we have $f(\Phi_x \bigcup \{j\}) - f(\Phi_x) \geq f(\Phi_y \bigcup \{j\}) - f(\Phi_y)$. Therefore, $f(\cdot)$ is submodular.

Finally, $f(\cdot)$ is also monotone as the more candidate leaf nodes are merged, the larger the expected proof size, which indicates $f(\Phi_x) \leq f(\Phi_y)$ for any $\Phi_x \subseteq \Phi_y \subseteq \Phi'$. □

A well known result is that for any objective function that is non-negative, submodular, and monotone, a greedy algorithm that iteratively selects the local optimal element at every step can output a solution with guaranteed approximation ratio of $1 - 1/e$, and no polynomial-time algorithm can achieve a better guarantee unless $P = NP$ [83].

We now detail the greedy algorithm for the merging decision in Algorithm 3. We first initialize the number of leaf nodes θ_i to $\phi_i - |\Phi_1|$, i.e., ϕ_i candidate nodes after $|\Phi_1|$ merging operations (Line 1). We then initialize Φ'_1 to empty set and the set of remaining decision variables Φ' to $\Phi \setminus (\Phi_0 \bigcup \Phi_1)$. We then iteratively make the

Algorithm 3: Minimizing Expected Proof Size

input : Candidate leaf nodes $C_{i,1}, \ldots, C_{i,\phi_i}$, Φ, Φ_0, Φ_1, and τ
output: Φ'_1 and Φ'_0

1 $\theta_i \leftarrow \phi_i - |\Phi_1|$;
2 $\Phi'_1 \leftarrow \emptyset$;
3 $\Phi' \leftarrow \Phi \setminus (\Phi_0 \bigcup \Phi_1)$;
4 **while** $\theta_i > \max(\tau, |\Phi_0| + 1)$ **do**
5 $\quad j^* = \arg\min_{j \in \Phi'} f(\Phi' \bigcup \{j\})$;
6 $\quad \Phi'_1 \leftarrow \Phi'_1 \bigcup \{j^*\}$;
7 $\quad \Phi' \leftarrow \Phi' \setminus \{j^*\}$;
8 $\quad \theta_i \leftarrow \theta_i - 1$;
9 **end**
10 $\Phi'_0 \leftarrow \Phi' \setminus \Phi'_1$;
11 **return** Φ'_1 and Φ'_0;

remaining merging decisions (Lines 4 to 9). In each iteration, we find $j^* \in \Phi'$ with the smallest $f(\Phi' \{j^*\})$ and move j^* from Φ' to Φ'_1. This process continuous until the number of leaf nodes θ_i reaches $\max(\tau, |\Phi_0| + 1)$. Finally, Φ'_1 and $\Phi'_0 = \Phi' \setminus \Phi'_1$ are output for constructing the leaf nodes for LKS-MHT T_i.

4.4.5.2 Formulation 2: Minimizing Maximal Size of Freshness Proof

Our second formulation seeks to minimize the maximal freshness proof size among all keys, i.e., $\max_{k \in \mathcal{K}}\{h_{k,i}\}$, under the constraint of the maximal number of leaf nodes. Note that this would require the data owner to keep track of $\{h_{k,i} | k \in \mathcal{K}\}$. Again let $h_{k,i}$ and $h_{k,i-1}$ be the number of leaf nodes that need be returned in response to queries $Q = (k, i)$ and $Q = (k, i-1)$, respectively, for all $k \in \mathcal{K}$. Recall that $\mathcal{K}_i \subseteq \mathcal{K}$ is the subset of keys that receive an update in interval i. It follows that $h_{k,i} = 1$ for all $k \in \mathcal{K}_i$. Since $h_{k,i} \geq 1$ for all $k \in \mathcal{K}$, we have

$$\max_{k \in \mathcal{K}}\{h_{k,i}\} = \max_{k \in \mathcal{K} \setminus \mathcal{K}_i}\{h_{k,i}\}. \tag{4.3}$$

Let $C_i^- = \{C_{i,j} | K_{i,j} \bigcap \mathcal{K}_i = \emptyset\}$, i.e., $C_{i,j}$ contains no key that receives an update in interval i. For every candidate leaf node $C_{i,j} \in C_i^-$, denote its maximum freshness

proof size in response to $Q = (k, i-1)$ and $Q = (k, i)$ by $m_{i-1,j} = \max_{k \in K_{i,j}}\{h_{k,i-1}\}$ and $m_{i,j} = \max_{k \in K_{i,j}}\{h_{k,i}\}$, respectively. It follows that

$$\begin{aligned} \max_{k \in \mathcal{K} \setminus \mathcal{K}_i} \{h_{k,i}\} &= \max_{C_{i,j} \in C_i^-} \{m_{i,j}\} \\ &= \max_{C_{i,j} \in C_i^-} \{m_{i-1,j} + \triangle m_j\}, \end{aligned} \tag{4.4}$$

where $\triangle m_j = m_{i,j} - m_{i-1,j}$ for all $C_{i,j} \in C_i^-$.

We now analyze the relationship between decision variables $b_1, \ldots, b_{\phi_i - 1}$ and $\max_{C_{i,j} \in C_i^-}\{m_{i-1,j} + \triangle m_j\}$. Similar to the case of Formulation 1, $\triangle m_j = 1$ if the candidate leaf node $C_{i,j}$ is merged with another and 0 otherwise. Again let $\Phi' = \Phi \setminus (\Phi_0 \bigcup \Phi_1)$ and $\{b_j | j \in \Phi'\}$ be the remaining decision variables that need be determined. Also let $\Phi'_1 = \{b_j = 1 | j \in \Phi'\}$ and $\Phi'_0 = \{b_j = 0 | j \in \Phi'\}$ be the subsets of decision variables set to one and zero, respectively. Given Φ'_1 and Φ_1, a candidate leaf node $C_{i,j}$ is merged with another one if either $j-1$ or $j \in \Phi'_1 \bigcup \Phi_1$. Let $\Pi = \{j | j-1 \in \Phi'_1 \bigcup \Phi_1 \vee j \in \Phi'_1 \bigcup \Phi_1 \wedge j \in \Phi\}$. We have

$$\max_{C_{i,j} \in C_i^-} \{m_{i,j}\} = \max(\{m_{i-1,j} + 1\}_{j \in \Pi}, \{m_{i-1,j}\}_{j \in \Phi \setminus \Pi}). \tag{4.5}$$

Let $g(\Phi'_1) = \max(\{m_{i-1,j} + 1\}_{j \in \Pi}, \{m_{i-1,j}\}_{j \in \Phi \setminus \Pi})$. We formulate the remaining merging decisions as the following optimization problem.

$$\begin{aligned} \text{minimize} \quad & g(\Phi'_1) \\ \text{subject to} \quad & \Phi'_1 \subseteq \Phi', \\ & \phi_i - |\Phi_1 \bigcup \Phi'_1| \leq \max(\tau, |\Phi_0| + 1), \\ & b_j = 0, \forall j \in \Phi_0 \bigcup \Phi'_0, \\ & b_j = 1, \forall j \in \Phi_1 \bigcup \Phi'_1. \end{aligned} \tag{4.6}$$

Theorem 10. *The objective function $g(\cdot)$ in Eq. (4.6) is non-negative, submodular, and monotone.*

Proof. First, we note that the objection function $g(\cdot)$ is non-negative as $\max(\{m_{i-1,j} + 1\}_{j \in \Pi}, \{m_{i-1,j}\}_{j \in \Phi \setminus \Pi}) \geq 0$ for all $j \in \Phi' \subseteq \Phi$.

Second, we show that $g : 2^{\Phi'} \to \mathbb{R}$ is submodular. Consider any two subsets Φ_x and Φ_y where $\Phi_x \subseteq \Phi_y \subset \Phi'$. There are two cases. First, if $\Phi_x = \Phi_y$, then $\Phi_x \bigcup \{j\} = \Phi_y \bigcup \{j\}$ for any $j \in \Phi'$. It follows that $g(\Phi_x \bigcup \{j\}) - g(\Phi_x) = g(\Phi_y \bigcup \{j\}) - g(\Phi_y)$. We now consider the second case where $\Phi_x \subset \Phi_y$. For any $j \in \Phi' \setminus \Phi_y$, i.e., C_j and C_{j+1} are two adjacent candidate leaf nodes, the objective function $g(\cdot)$ satisfies one of the following four cases.

- Case 1: if $\{j-1, j+1\} \subseteq \Phi_y \setminus \Phi_x$, then both C_j and C_{j+1} are merged with other candidate leaf nodes in Φ_y, but none of them has been merged with another in Φ_x, which indicates that $g(\Phi_y \bigcup \{j\}) - g(\Phi_y) = 0$ and $g(\Phi_x \bigcup \{j\}) - g(\Phi_x) > 0$. Thus, for any $j \in \Phi' \setminus \Phi_y$ $g(\Phi_x \bigcup \{j\}) - g(\Phi_x) > g(\Phi_y \bigcup \{j\}) - g(\Phi_y) = 0$.
- Case 2: if $j - 1 \in \Phi_y \setminus \Phi_x$ and $j + 1 \notin \Phi_y \setminus \Phi_x$, then C_j has been merged with another one in Φ_y resulting in smaller return from adding j. Thus, for any $j \in \Phi' \setminus \Phi_y$ $g(\Phi_x \bigcup \{j\}) - g(\Phi_x) > g(\Phi_y \bigcup \{j\}) - g(\Phi_y)$.
- Case 3: if $j - 1 \notin \Phi_y \setminus \Phi_x$ and $j + 1 \in \Phi_y \setminus \Phi_x$, then it is symmetric to Case 2, which leads to the same conclusion that for any $j \in \Phi' \setminus \Phi_y$, $g(\Phi_x \bigcup \{j\}) - g(\Phi_x) > g(\Phi_y \bigcup \{j\}) - g(\Phi_y)$.
- Case 4: if $j - 1 \notin \Phi_y \setminus \Phi_x$ and $j + 1 \notin \Phi_y \setminus \Phi_x$, then neither C_j nor C_{j+1} have been merged with another one in Φ_y resulting in the same return from adding j. Thus, for any $j \in \Phi' \setminus \Phi_y$ we have $g(\Phi_x \bigcup \{j\}) - g(\Phi_x) = g(\Phi_y \bigcup \{j\}) - g(\Phi_y)$.

To sum up, for any $\Phi_x \subseteq \Phi_y \subset \Phi'$ and $j \in \Phi' \setminus \Phi_y$, we have $g(\Phi_x \bigcup \{j\}) - g(\Phi_x) \geq g(\Phi_y \bigcup \{j\}) - g(\Phi_y)$. Therefore, $g(\cdot)$ is submodular.

Finally, $g(\cdot)$ is also monotone as the more candidate leaf nodes are merged, the larger the maximal proof size, which indicates $g(\Phi_x) \leq g(\Phi_y)$ for any $\Phi_x \subseteq \Phi_y \subseteq \Phi'$. $\square$

We now introduce an efficient greedy algorithm to solve the above optimization problem. While choosing the local optimal with the smallest $g(\cdot)$ can lead to an efficient greedy algorithm with guaranteed approximation ratio as in the first formulation, we notice that there may be multiple choices with the same minimal $g(\cdot)$ in each step. We

therefore further prioritize the merging decision that involves the new candidate leaf nodes with the smallest key block size.

We detail the procedure for determining the merging decisions in Algorithm 4. Specifically, we first initialize the number of merging decisions needed θ_i to $\phi_i - |\Phi_1|$, output Φ_1' to the empty set, and remaining merging decisions Φ' to $\Phi \setminus (\Phi_0 \bigcup \Phi_1)$ (Lines 1 to 3). We then define a variable flag_j for each candidate leaf node $C_{i,j}$ to indicate whether it has been merged with another and initiate $\{\mathsf{flag}_j\}_{j=1}^{\phi_i}$ based on predetermined merging decisions Φ_1 (Lines 4 to 10). We then define three temporary variables TempObj, TempSize, and TempIndex to store the local merging choice, affected new candidate leaf node's key block size, and corresponding value of the objective function, respectively (Lines 11 to 13). In the subsequent While loop (Lines 14 to 35), we iteratively select merging decisions until the terminal condition is met. Specifically, in each iteration, we check each of the remain merging decisions $j \in \Phi'$ to find the one with the smallest objective function value $g(\Phi_1' \bigcup j)$ (Lines 16 to 22). If there are multiple ones with the same smallest objective function value, we then break the tie by choosing the one that introduces the smallest key block size (Lines 23 to 32).

Algorithm 4: Minimizing Maximum Proof Size

input : Candidate leaf nodes $C_{i,1}, \ldots, C_{i,\phi_i}$, Φ, Φ_0, Φ_1, τ, and $\{h_{k,i-1} | k \in \mathcal{K}\}$
output: Φ'_1 and Φ'_0

$\theta_i \leftarrow \phi_i - |\Phi_1|$; $\Phi'_1 \leftarrow \emptyset$; $\Phi' \leftarrow \Phi \setminus (\Phi_0 \bigcup \Phi_1)$;
forall $j \in \{1, \ldots, \phi_i\}$ **do**
    $\mathsf{flag}_j \leftarrow \mathsf{false}$;
forall $j \in \Phi_1$ **do**
    if $\mathsf{flag}_j = \mathsf{false}$ **then**
        $\mathsf{flag}_j \leftarrow \mathsf{true}$;
    if $\mathsf{flag}_{j+1} = \mathsf{false}$ **then**
        $\mathsf{flag}_{j+1} \leftarrow \mathsf{true}$;
TempIndex $\leftarrow$ null; TempObj $\leftarrow \infty$; TempSize $\leftarrow 0$;
while $\theta_i > \max(\tau, |\Phi_0| + 1)$ **do**
    forall $j \in \Phi'$ **do**
        if $g(\Phi'_1 \bigcup \{j\} <$ TempObj$)$ **then**
            TempIndex $\leftarrow j$;
            if $\mathsf{flag}_j = \mathsf{false}$ **then**
                TempSize $\leftarrow |K_{i,j}|$;
            if $\mathsf{flag}_{j+1} = \mathsf{false}$ **then**
                TempSize $\leftarrow$ TempSize $+ |K_{i,j+1}|$;
            TempObj $\leftarrow g(\Phi'_1 \bigcup \{j\})$;
        else if $g(\Phi'_1 \bigcup \{j\} =$ TempObj$)$ **then**
            TempSize$' \leftarrow 0$;
            if $\mathsf{flag}_j = \mathsf{false}$ **then**
                TempSize$' \leftarrow |K_{i,j}|$;
            if $\mathsf{flag}_{j+1} = \mathsf{false}$ **then**
                TempSize$' \leftarrow$ TempSize$' + |K_{i,j+1}|$;
            if TempSize$'$ < TempSize **then**
                TempIndex $\leftarrow j$;
                TempSize $\leftarrow$ TempSize$'$;
                TempObj $\leftarrow g(\Phi'_1 \bigcup \{j\})$;
    $\Phi'_1 \leftarrow \Phi'_1 \bigcup \{$TempIndex$\}$;
    $\Phi' \leftarrow \Phi' \setminus \{$TempIndex$\}$;
    $\theta_i \leftarrow \theta_i - 1$;
$\Phi'_0 \leftarrow \Phi' \setminus \Phi'_1$;
return Φ'_1 and Φ'_0;

4.4.6 Point Query Processing

We now detail the procedure of KV-Fresh for point queries, which consists of three phases: *update preprocessing*, *query processing*, and *query-result verification*. We assume that the data owner has a public/private key pair that supports batch verification of digital signatures such as RSA [110].

Update Preprocessing. Assume that the data owner receives data records $\{\langle v_k^i, t_k^i\rangle | k \in \mathcal{K}_i\}$ in each interval i for $i = 1, 2, \ldots$. At the end of each interval i, the data owner generates the leaf nodes $L_{i,1}, \ldots, L_{i,\theta_i}$ according to the procedures presented in Section 4.4.4 if $i = 1$ or Section 4.4.5 otherwise. The data owner then constructs an LKS-MHT T_i over $L_{i,1}, \ldots, L_{i,\theta_i}$. Let (n, e) and d be the data owner's RSA public/private key pair and R_i the root of T_i. The data owner computes

$$s_i = H(i||R_i)^d \mod n.$$

Finally, the data owner sends all the leaf nodes $L_{i,1}, \ldots, L_{i,\theta_i}$ and its signature s_i to the cloud server, whereby the cloud server can compute all the intermediate nodes and root of T_i.

Query Processing. Assume that a data user issues a GET query $Q(k, t_q)$ asking for the most recent data record for key k as of the end of interval q_1. Also assume that v_k^i is the most recent update for key k received at time t_k^i in interval i, where $i \leq q_1$.

Given $T_1, \ldots, T_{q_1}$, the cloud server constructs the query result from the leaf nodes that containing key k in a subset of LKS-MHTs to prove the freshness of v_k^i. Specifically, the cloud server first finds the leaf node L_{q_1,j_1} in LKS-MHT T_{q_1} such that $k \in K_{q_1,j_1}$. If $i = q_1$, then we have $L_{q_1,j_1} = \langle k, v_k^i, t_k^i\rangle$. Otherwise, $L_{q_1,j_1} = \langle K_{q_1,j_1}, \gamma_{q_1,j_1}\rangle$. Specifically, for every $x = 1, 2, \ldots$, the cloud server finds the leaf node L_{q_x,j_x} in LKS-MHT T_{q_x} such that $k \in K_{q_x,j_x}$. It follows that $L_{q_x,j_x} = \langle k, v_k^i, t_k^i\rangle$ if $q_x = i$ and $\langle K_{q_x,j_x}, \gamma_{q_x,j_x}\rangle$ otherwise. The cloud server returns

$$\mathsf{R}_x = \langle q_x, L_{q_x,j_x}, \mathcal{A}(R_{q_x}|L_{q_x,j_x}), s_{q_x}\rangle, \tag{4.7}$$

as a partial query result, where R_{q_x} is the root of LKS-MHT T_{q_x}, and $\mathcal{A}(R_{q_x}|L_{q_x,j_x})$ is the set of internal nodes in T_{q_x} needed for computing root R_{q_x} from leaf node L_{q_x,j_x}. If $q_x > i$, then the cloud server set $q_{x+1} = \gamma_{q_x,j_x}$ and repeat the above process until $q_x = i$, i.e., the most recent update for key k received in interval i has been returned.

Query-Result Verification. Assume that the user has received the query result in the form of $\mathsf{R} = \langle \mathsf{R}_1, \ldots, \mathsf{R}_r \rangle$, where $\mathsf{R}_x = \langle q_x, L_{q_x,j_x}, \mathcal{A}(R_{q_x}|L_{q_x,j_x}), s_{q_x} \rangle$, for all $1 \leq x \leq r$. The data user first verifies the integrity of the query result. Specifically, for every $x = 1, \ldots, r$, the user first computes R_{q_x} from L_{q_x,j_x} using $\mathcal{A}(R_{q_x}|L_{q_x,j_x})$. It then verifies all r signatures in batch by checking whether

$$(\prod_{x=1}^{r} s_{q_x})^e \stackrel{?}{=} \prod_{x=1}^{r} H(q_x||R_{q_x}) \mod n,$$

where (n, e) is the data owner's RSA public key. If so, the user considers the query result authentic.

The data user also proceeds to verify the freshness of the query result using the interval indexes embedded in the returned leaf nodes. Assume that $q_1 > \cdots > q_s$. The user first checks if $q_s = q_1$, as the cloud server should always return one leaf node for the queried interval q_1. If so, the user further checks whether $q_{x+1} = \gamma_{q_x,j_x}$ for all $x = 1, \ldots, s-1$. Finally, the user verifies whether leaf node L_{q_x,j_x} contains the updated value v_k^i and timestamp t_k^i. If so, the user considers the query result fresh and accepts v_k^i as the most recent.

4.4.7 Range Query Processing

We now discuss how to extend the above solution for point query into range query. A straightforward solution is to convert any range query into multiple point queries with each corresponding to one unique queried key. Under this approach, the proof size is approximately linear to the size of query range, which would incur significant communication overhead when the query range is large. Our key observation is that the point query responses with respect to adjacent queried keys have large overlap and can be merged to significantly reduce the communication overhead. In

what follows, we detail the procedures of query processing and query-result verification, as that of update preprocessing is identical to the case of point query.

Query Processing. Assume that the cloud server receives a GET range query $Q([l,r],t_q)$ asking for the most recent data record for every key $k \in [l,r]$ as of the end of interval q. Also assume that v_k is the most recent update received at time t_k^i in interval i_k, where $i_k \leq q$ for all $k \in [l,r]$).

The cloud server first generates a point query result for every queried key $k \in [l,r]$. Let $\mathsf{R}^k = \{\mathsf{R}_1^k, \ldots, \mathsf{R}_{r_k}^k\}$ be the query result for each queried key $k \in [l,r]$, where r_k is the number of partial query results and

$$\mathsf{R}_x^k = \langle q_x^k, L_{q_x,j_x}^k, \mathcal{A}(R_{q_x^k}^k | L_{q_x,j_x}^k), s_{q_x^k} \rangle, \tag{4.8}$$

for all $1 \leq x \leq r_k$. It is easy to see that $q_1^k = q$ for all $l \leq k \leq r$ as the query result for every queried key must contain the information about interval q.

Given all the partial query results $\{\mathsf{R}_x^k | l \leq k \leq r, 1 \leq x \leq r_k\}$, the cloud server then constructs the final query result in two steps. First, the cloud server sorts $\{\mathsf{R}_x^k | l \leq k \leq r\}$ first according to interval index q_x^k and then key k such that partial query results for the adjacent keys and the same interval appears next to each other. The cloud server then identifies and eliminates duplicate partial query results for different keys for the same interval. Second, the cloud server merges all the partial query results into one for every interval that appears in $\{\mathsf{R}_x^k | l \leq k \leq r\}$. Specifically, let $i^* = \min_{k \in [l,r]}\{i_k\}$ be the earliest interval with the most recent update for any queried key. For every interval $j \in [i^*, q]$ with at least one partial query result, the cloud server constructs an aggregated partial query result as follows. Let $\mathcal{K}_j^{[l,r]} \in [l,r]$ be the subset of keys that have partial query results for interval j. For each $k \in \mathcal{K}^{[l,r]}$, let its partial query result for interval j be

$$\mathsf{R}^k = \langle j, L_j^k, \mathcal{A}(R^k | L_j^k), s_j \rangle, \tag{4.9}$$

where we omit a part of the subscript to simplify the notation. We can see that $\{L_j^k\}_{k \in [l,r]}$ are a subset of LKS-MHT T_j's leaf nodes. The cloud server constructs an

aggregate query result for interval j as

$$\mathsf{R}_j^{[l,r]} = \langle j, \{L_j^k | k \in \mathcal{K}_j^{[l,r]}\}, \mathcal{A}(R^k | \{L_j^k | k \in \mathcal{K}_j^{[l,r]}\}), s_j \rangle, \tag{4.10}$$

where $\mathcal{A}(R^k | \{L_j^k | k \in \mathcal{K}_j^{[l,r]}\})$ is the union of the subsets of internal nodes of LKS-MHT T_j needed to compute the root R^k from L_j^k for all $k \in \mathcal{K}^{[l,r]}$.

Query-Result Verification. The verification of a range query result is essentially the same as verifying multiple point query results. In particular, the only difference between the query processing in the two cases is that the cloud server eliminates the duplicated information among multiple point query results, so all the information needed for verifying the integrity and freshness of individual point query results are included in the range query result. We omit the details here due to overlap.

4.5 Performance Evaluation

In this section, we evaluate the performance of KV-Fresh via extensive simulation studies using a real dataset.

4.5.1 Dataset

We create a synthetic dataset from a TrueFax real-time currency conversion dataset [111] that includes tick-by-tick historical conversion rates for 16 major currency pairs with fractional pip spreads in millisecond detail. For our purpose, we take the currency conversion rate from EUR to USD from 12:00 am (GMT), January 2nd, 2019 to 03:46:40 pm (GMT) January 3rd, 2019. We divide the time period into 10,000 segments of 10 seconds. We treat the segment indexes as keys and the conversion rates as the updates. Our synthetic dataset consists of 10,000 keys for a period of 10 seconds, and on average 131.55 keys receive updates for every 10 ms.

4.5.2 Simulation Settings

We implement KV-Fresh in Python and test it on a desktop with i7-6700 CPU, 16GB RAM, and 64-bit Win10 operating system. We adopt the SHA-256 for the

Table 4.1: Default Simulation Settings

Para.	Val.	Description.
ϵ	10 ms	The interval size
$\|\mathcal{K}\|$	10,000	The number of keys
m	1,000	The number of intervals
τ	1024	The maximal number of key blocks
$\|H(\cdot)\|$	256	The length of hash
$\|s_i\|$	1024	The length of data owner's signature

cryptographic hash function and RSA for digital signature. Table 4.1 summarizes our default settings unless mentioned otherwise.

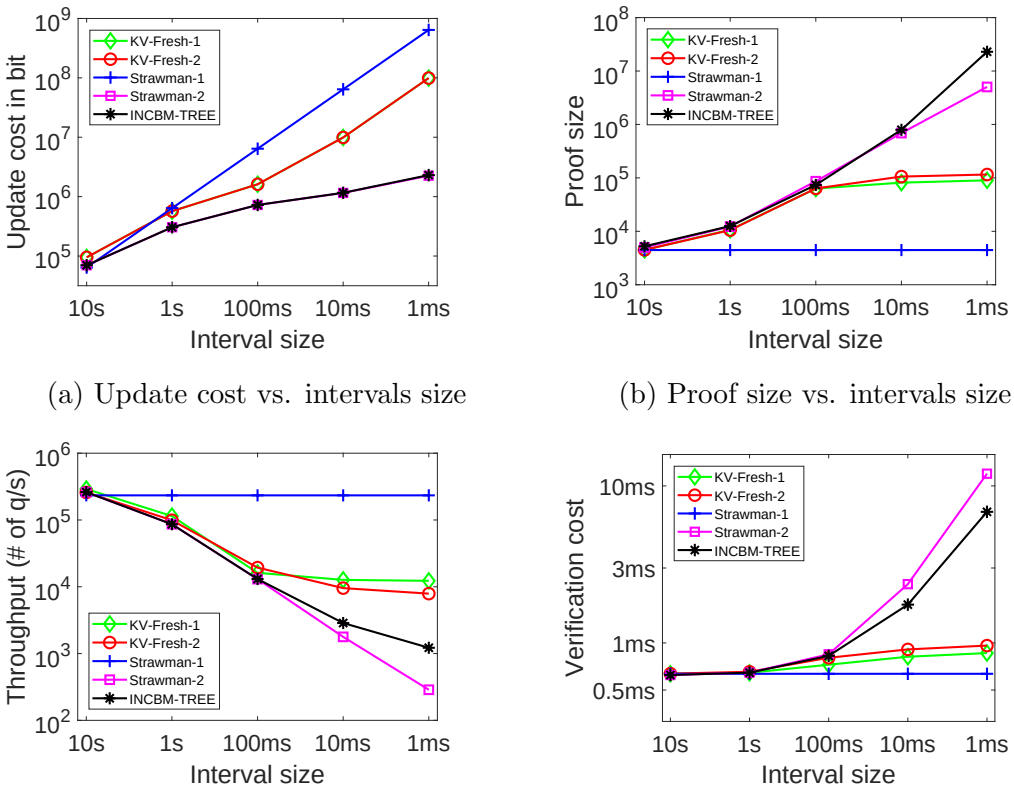
For point query, we compare KV-Fresh with the state-of-art solution INCBM-TREE [101] as well as the Strawman-1 and Strawman-2 approaches discussed in Section 4.4.1 using four performance metrics: (1) *update cost* which is the number of extra bits per second transmitted from the data owner to cloud server, (2) *proof size* which is the number of extra bits needed for proving the integrity and freshness for a query result, (3) *throughput* which is the number of queries processed by the cloud server per second, and (4) *verification time* which is the time needed for verifying a returned query result by the user.

4.5.3 Simulation Results for Point Queries

We now report our simulation results for point queries where every point in the following figures represents the average over 10,000 runs each with a distinct random seed. We refer to the two formulations discussed in Sections 4.4.5.1 and 4.4.5.2 as KV-Fresh-1 and KV-Fresh-2, respectively.

4.5.3.1 The Impact of Interval Size

Fig. 4.6a compares the update cost under Strawman-1, Strawman-2, INCBM-TREE, KV-Fresh-1, and KV-Fresh-2 with interval size varying from 10 s to 1 ms, respectively. As we can see, the update cost per second increases as the interval sizes decreases under all mechanisms. This is expected, as the number of intervals is inversely proportional to the interval size. Among the five mechanisms, Strawman-1 has the



(a) Update cost vs. intervals size

(b) Proof size vs. intervals size

(c) Throughput vs. intervals size

(d) Verification cost vs. intervals size

Figure 4.6: Comparison of KV-Fresh, Strawman-1, Strawman-2, and INCBM-TREE with interval size varying from 10s to 1ms.

highest update cost when the interval size is smaller than 1 s, as the data owner needs to send the most recent key-value record for every key in every interval. Strawman-2 and INCBM-TREE have the lowest update cost, as the data owner only sends keys with updates under both mechanisms. The update costs of KV-Fresh-1 and KV-Fresh-2 fall in the middle and increase much slower than that of Strawman-1. This is anticipated, as both KV-Fresh-1 and KV-Fresh-2 require the data owner to send only updated key-value records and key block information with no update for every interval. Moreover, when the interval size is 1 ms, both KV-Fresh-1 and KV-Fresh-2 incur an update cost of approximately 10^8 bits per second. In other words, a 100-Mbps link between the data owner and the cloud server suffices to support a key space of 10,000 keys, which makes KV-Fresh very practical.

Fig. 4.6b shows the impact of interval size on the proof size of Strawman-1, Strawman-2, INCBM-TREE, KV-Fresh-1, and KV-Fresh-2. The proof size of Strawman-1 is not affected by the interval size and stays at 4460 bits. The proof sizes of the other four mechanisms all increase as the interval size decreases. Among the them, the proof sizes of Strawman-2 and INCBM-TREE grow the fastest and are approximately inversely proportional to the interval size. The reason is that the data owner needs to prove that there is no update in every interval after the most recent update under the both mechanisms. While INCBM-TREE employs a Bloom filter for efficient proof of no update, every Bloom filter covers only a constant number of intervals. In contrast, the proof sizes under KV-Fresh-1 and KV-Fresh-2 grow much slower as the interval size decreases, because both KV-Fresh-1 and KV-Fresh-2 allow the cloud server to skip potentially many intervals in the freshness proof. We can also see that the proof size of KV-Fresh-1 is slightly lower than that of KV-Fresh-2, which is anticipated as KV-Fresh-1 aims to minimizing the expected size of freshness proof and the proof size in Fig. 4.6b is the average over 10,000 runs. In addition, we can see that KV-Fresh outperforms INCBM-TREE by a large margin when the interval size is small. For example, when the interval size is 1 ms, the proof sizes under KV-Fresh-1 and KV-Fresh-2 are approximately 90 Kb and 115 Kb, respectively, which are less than 0.4% and 0.5% of the 22.9 Mb under INCBM-TREE, respectively.

Fig. 4.6c compares the throughput under Strawman-1, Strawman-2, INCBM-TREE, KV-Fresh-1, and KV-Fresh-2. We can see that the throughput under Strawman-1 is the highest and not affected by the change in interval size. Among the other four, the throughput of Strawman-2 is the smallest, followed by INCBM-TREE. The reason is that the smaller the interval size, the more intervals after the most recent update on average, the more intervals the cloud server needs to process under Strawman-2 and INCBM-TREE, and vice versa. In contrast, the throughput of KV-Fresh-1 and KV-Fresh-2 initially decline as the interval size decreases from 10 s to 10 ms and then become stable or decrease slightly as the interval size decreases from 10 ms to 1 ms. The reason for the initial decline is that when the interval size is large, most of the keys

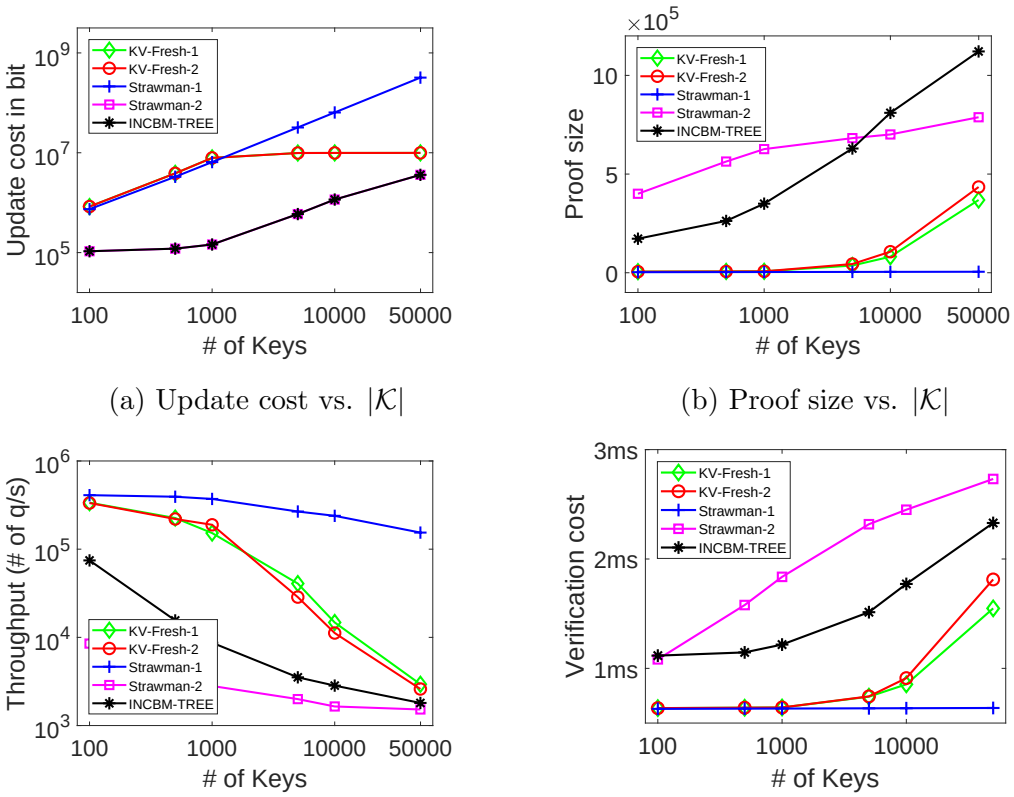
have updates in every interval, and the merging constraint is determined by $|\Phi_0|$ instead of τ, which results in excessive merging operations and more intervals that the cloud server needs to check. As the interval size further decreases, fewer and fewer keys have updates in each interval, which results in fewer merging operations and thus fewer intervals the cloud server needs to check. Moreover, KV-Fresh-1 outperforms KV-Fresh-2 with higher average throughput due to its merging decision policy, which aims to minimize the expected proof size. Generally speaking, in comparison with Strawman-2 and INCBM-TREE, both KV-Fresh-1 and KV-Fresh-2 have similar throughput when the interval size is large while outperforming Strawman-2 and INCBM-TREE by large margins when the interval size is small. For example, when the interval size is 1 ms, KV-Fresh-1 achieves 9.05 and 41.75 times higher throughput than INCBM-TREE and Strawman-2, respectively.

Fig. 4.6d compares the verification cost of the five mechanisms under different interval sizes. As we can see, the verification cost of Strawman-1 remains at 0.6357ms and is not affected by the change in interval size. The verification cost increases as the interval size decreases under all the other four mechanisms. Among them, KV-Fresh-1 and KV-Fresh-2 both outperform INCBM-TREE and Strawman-2 by large margins. The reason is that fewer leaf nodes need be returned under either KV-Fresh-1 or KV-Fresh-1 than both INCBM-TREE and Strawman-2. For example, when interval size is 1 ms, it takes 0.86 ms and 0.96 ms to verify a query result under KV-Fresh-1 and KV-Fresh-2, respectively, while Strawman-2 and INCBM-TREE require 11.96 ms and 6.84 ms, respectively.

These results demonstrate the significant advantages of KV-Fresh over other two mechanisms.

4.5.3.2 The Impact of the Number of Keys

Figs. 4.7a to 4.7d compare the performance of KV-Fresh-1, KV-Fresh-2, Strawman-1, Strawman-2 and INCBM-TREE with $|\mathcal{K}|$, i.e., the total number of keys, varying from 100 to 50,000. As we can see from Fig. 4.7a, the update costs of all schemes increase



(a) Update cost vs. $|\mathcal{K}|$

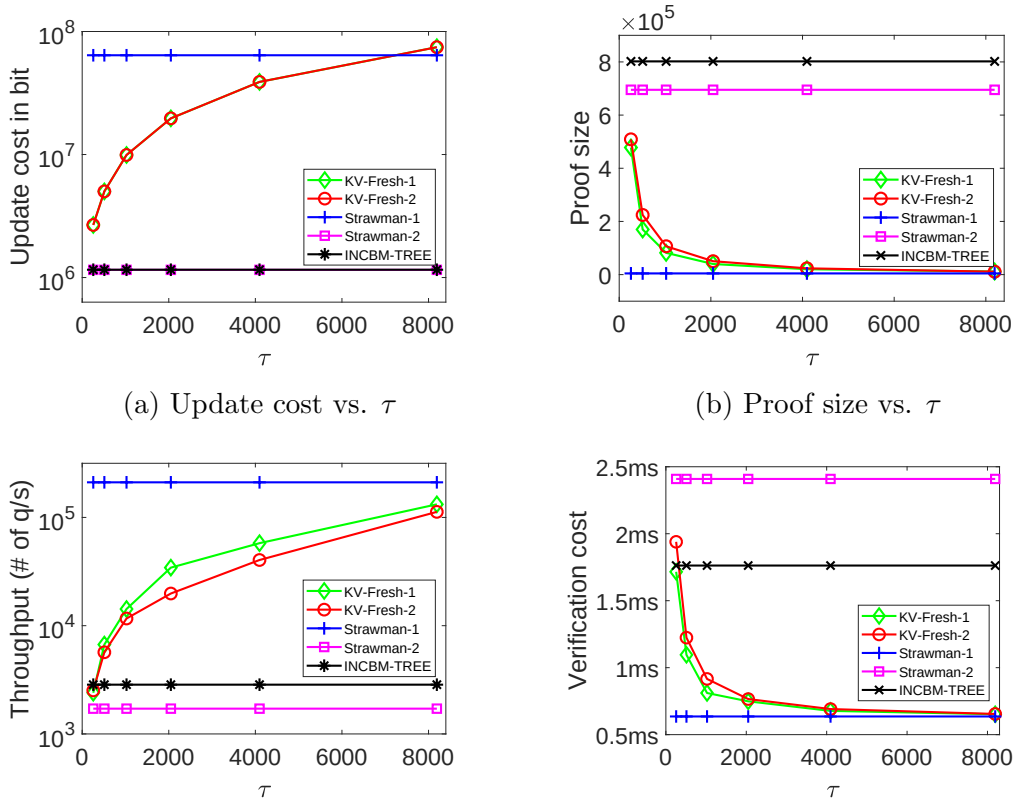
(b) Proof size vs. $|\mathcal{K}|$

(c) Throughput vs. $|\mathcal{K}|$

(d) Verification cost vs. $|\mathcal{K}|$

Figure 4.7: Comparison of KV-Fresh, Strawman-1, Strawman-2, and INCBM-TREE with $|\mathcal{K}|$ varying from 100 to 50,000.

as the number of keys increase, which is anticipated. Moreover, the update cost of KV-Fresh-1 and KV-Fresh-2 are lower than that of Strawman-1 by a larger margin but higher than that of Strawman-2 and INCBM-TREE. More importantly, even when the $|\mathcal{K}|$ is 50,000, the update costs of KV-Fresh-1 and KV-Fresh-2 are both approximately 3.9×10^7 bits per second, which is very practical for 10-ms interval. From Fig. 4.7b, we can see that the proof sizes under all mechanisms increase as $|\mathcal{K}|$ increases, as a larger $|\mathcal{K}|$ leads to a deeper MHT. Moreover, as $|\mathcal{K}|$ increases from 100 to 50,000, the proof sizes under KV-Fresh-1 and KV-Fresh-2 are always significantly smaller than those under Strawman-2 and INCBM-TREE. Similarly, Figs. 4.7c and 4.7d show that both KV-Fresh-1 and KV-Fresh-2 achieve much higher throughput and lower verification cost than Strawman-2 and INCBM-TREE because fewer leaf nodes need be returned



(a) Update cost vs. τ

(b) Proof size vs. τ

(c) Throughput vs.τ

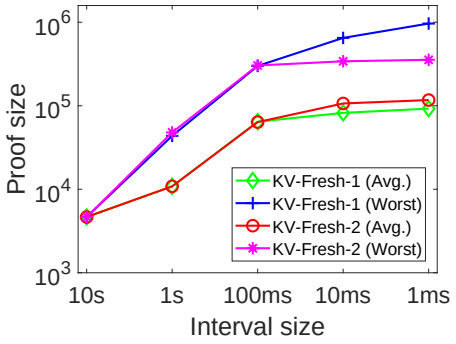
(d) Verification cost vs. τ

Figure 4.8: Comparison of KV-Fresh, Strawman-1, Strawman-2, and INCBM-TREE with τ varying from 256 to 10,000.

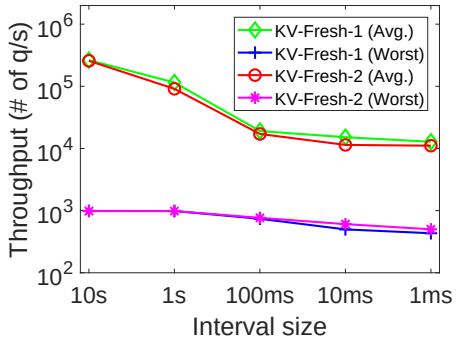
under KV-Fresh-1 and KV-Fresh-2 than the other two.

4.5.3.3 The Impact of τ

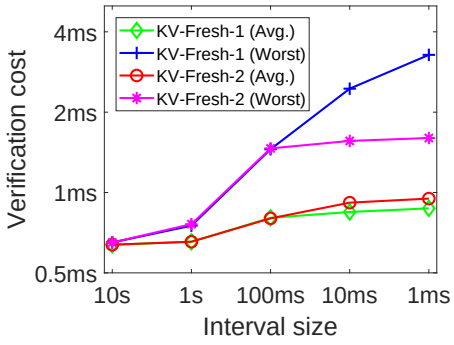
Figs. 4.8a to 4.8d show the performance of KV-Fresh-1 and KV-Fresh-2 with τ varying from 256 to 8192, where the performance of Strawman-1, Strawman-2, and INCBM-TREE are not affected by τ and only plotted for reference. Generally speaking, the larger τ, the higher the update cost, the smaller proof size, the higher throughput, the smaller verification cost for both KV-Fresh-1 and KV-Fresh-2, and vice versa. In addition, the update cost, proof size, throughput, and verification cost under KV-Fresh-1 and KV-Fresh-2 are almost always between those under Strawman-1 and those under Strawman-2 and INCBM-TREE, which is expected. While KV-Fresh-1 and



(a) Proof size vs. interval size



(b) Throughput vs. interval size



(c) Verification cost vs. interval size

Figure 4.9: Comparison of KV-Fresh-1 and KV-Fresh-2 with interval size varying from 10s to 1ms.

KV-Fresh-2 incur higher update cost than Strawman-2 and INCBM-TREE, they incur much lower communication cost between the cloud server and the user and smaller verification cost at the user. Moreover, while update only happens between the data owner and the cloud server, the cloud server needs to serve potentially many users at the same time.

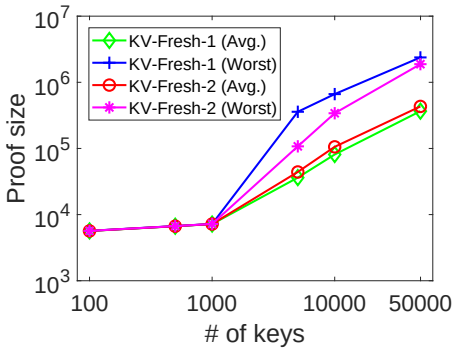
4.5.4 Comparison between KV-Fresh-1 and KV-Fresh-2

We now compare KV-Fresh-1 and KV-Fresh-2 in terms of their average and worst-case performance.

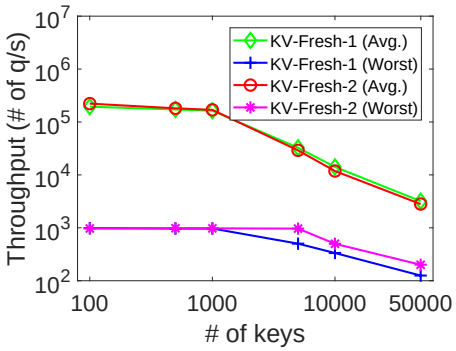
Fig. 4.9a and Fig. 4.9c compare the performance of KV-Fresh-1 and KV-Fresh-2 with interval size varying from 10 s to 1 ms, where KV-Fresh-1 (Avg.) and KV-Fresh-2

(Avg.) represent the average results of 10,000 runs and KV-Fresh-1 (Worst) and KV-Fresh-2 (Worst) represent the worst case among the 10,000 runs under KV-Fresh-1 and KV-Fresh-2, respectively. As we can see from Fig. 4.9a, as the interval size decreases, both the average and the largest proof sizes increase under both KV-Fresh-1 and KV-Fresh-2, which is expected. More importantly, KV-Fresh-1 achieves smaller average proof size but larger proof size under the worst case. The reason is that KV-Fresh-1 and KV-Fresh-2 are designed to minimize the expected and maximum proof sizes, respectively. Fig. 4.9b shows that as the interval size increases, both the average and maximum proof sizes initially decrease followed by stable or decrease slightly due to the same reason in Fig. 4.6c. We also observe that KV-Fresh-1 achieves higher average throughout but lower worst-case throughput than KV-Fresh-2. From Fig. 4.9c, we can see that KV-Fresh-2 incurs a slightly higher average verification cost than KV-Fresh-1 for the same reason. More importantly, the worst-case verification cost under KV-Fresh-2 is significantly lower than that of KV-Fresh-1. Moreover, we can see that the gap between the average and worst-case verification costs grows as the interval size decreases. The reason is that when the interval size is large, many keys receive updates in each interval on average, and the terminal condition for merging is mainly determined by τ, so there are very few merging opportunities to demonstrate the difference between KV-Fresh-1 and KV-Fresh-2. As the interval size decreases, the terminal condition is gradually determined by τ, and different merging decisions have large impact on the average and worst-case verification costs, which leads to the increased gap between the two mechanisms.

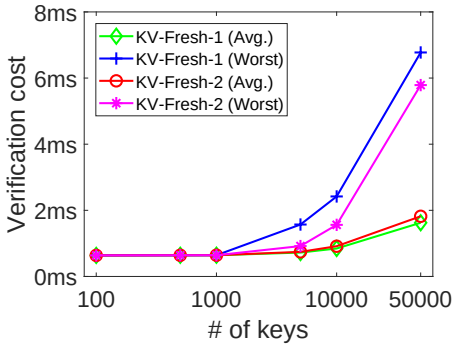
Fig. 4.10a and Fig. 4.10c compare the average and worst-case performance of KV-Fresh-1 and KV-Fresh-2 with $|\mathcal{K}|$ varying from 100 to 50,000. Generally speaking, the larger $|\mathcal{K}|$, the larger proof size, the lower throughout, and the higher verification cost for both the average and worst-case under the two mechanisms. Moreover, KV-Fresh-1 outperforms KV-Fresh-2 in terms of average proof size, throughout and verification cost, while KV-Fresh-2 has better worst-case performance. In addition,



(a) Proof size vs. $|\mathcal{K}|$



(b) Throughput vs. $|\mathcal{K}|$

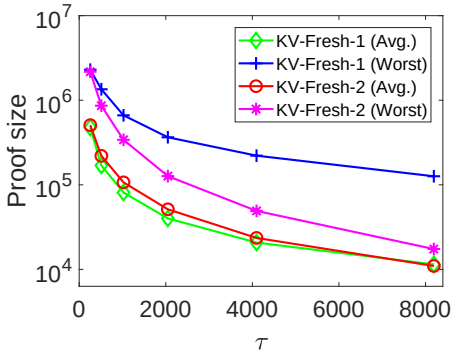


(c) Verification cost vs. $|\mathcal{K}|$

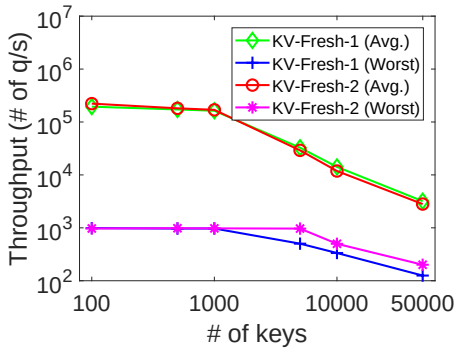
Figure 4.10: Comparison of KV-Fresh-1 and KV-Fresh-2 with $|\mathcal{K}|$ varying from 100 to 50,000.

we can see from Fig. 4.10c that the gap between the average and the worst-case performance increases as $|\mathcal{K}|$ increase from 100 to 50,000. For example, the difference between KV-Fresh-1 (Avg.) and KV-Fresh-1 (Worst) grows from 0.84 ms to 5.1 ms when the $|\mathcal{K}|$ increases from 5,000 to 50,000.

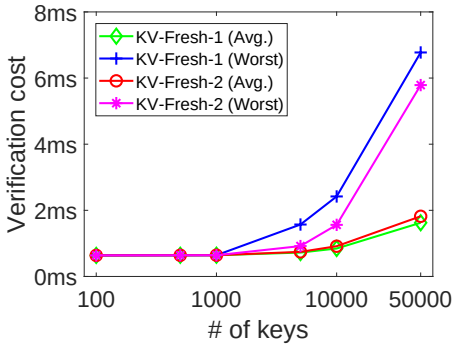
Figs. 4.11a to 4.11c compare the average and worst-case performance of KV-Fresh-1 and KV-Fresh-2 with with τ varying from 256 to 8192. Generally speaking, the larger τ, the higher update cost, the smaller proof size, the higher throughput, the smaller verification cost under both KV-Fresh-1 and KV-Fresh-2, and vice versa. In addition, KV-Fresh-1 (Avg.) always outperforms KV-Fresh-2 (Avg.) while KV-Fresh-2 (Worst) always outperforms KV-Fresh-1 (Worst), which are expected.



(a) Proof size vs. τ



(b) Throughput vs. τ



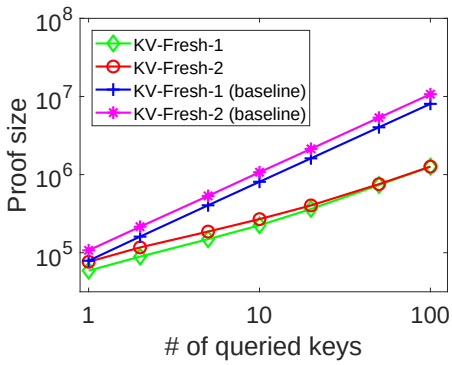
(c) Verification cost vs. τ

Figure 4.11: Comparison of KV-Fresh-1 and KV-Fresh-2 with tau varying from 256 to 8192.

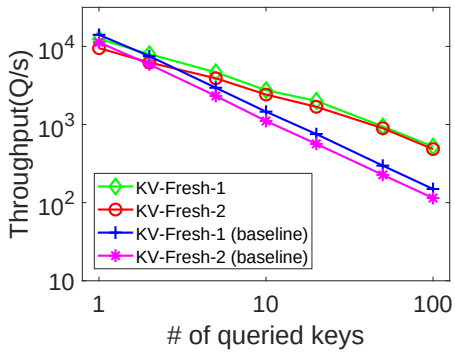
4.5.5 Simulation Results for Range Queries

We now report the simulation results for KV-Fresh on range queries. Since INCBM-TREE [101] is not directly applicable to range query, we compare KV-Fresh with a baseline solution, referred to as *KV-Fresh-baseline*, which processes a range query as multiple independent points as in KV-Fresh. For the performance evaluation, we still use the metrics *proof size*, *throughput*, and *verification time* but omit the metric *update cost* as KV-Fresh and KV-Fresh-baseline share the same update preprocessing procedure.

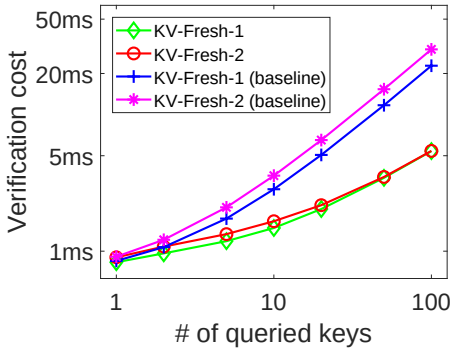
Fig. 4.12a shows the impact of the size of query range on the proof size under KV-Fresh-1, KV-Fresh-2, KV-Fresh-1-baseline, and KV-Fresh-2-baseline. We can see that the proof size increases as the number of queried keys increases under all four



(a) Proof size vs. size of query range



(b) Throughput vs. size of query range



(c) Verification cost vs. size of query range

Figure 4.12: Comparison of KV-Fresh-1 and KV-Fresh-2 with the size of query range varying from 1 to 100.

mechanisms, which is expected. Moreover, KV-Fresh-1 and KV-Fresh-2 both outperform corresponding KV-Fresh-1-baseline and KV-Fresh-2-baseline with a large margin. The reason is that the two baseline solutions treat a range query as multiple independent point queries for which the query results have large overlap. In contrast, both KV-Fresh-1 and KV-Fresh-2 eliminate such redundancy in the query result, resulting in significant reduction in the freshness proof size and thus higher communication and computation efficiency. In addition, the average proof size under KV-Fresh-1 is slightly lower than that under KV-Fresh-2 due to the same reason discussed in section 4.5.4

Fig. 4.12b compares the throughput of KV-Fresh-1, KV-Fresh-2, KV-Fresh-1-baseline, and KV-Fresh-2-baseline with the number of queried keys varying from 1 to 100. We can see that the throughput under all four mechanisms decreases as the

number of queried keys increase, which is expected as it takes longer time to process a range query with a larger query range size. Moreover, both KV-Fresh-1 and KV-Fresh-2 outperform corresponding KV-Fresh-1-baseline and KV-Fresh-2-baseline, especially when the size of query range is large, as they both treat a range query as a whole instead of multiple independent point queries. For example, when the size of query range is 100, KV-Fresh-1 can process 522 range queries in one second, while KV-Fresh-1-baseline can only process 149 range queries.

Fig. 4.12c shows the verification cost of KV-Fresh-1, KV-Fresh-2, KV-Fresh-1-baseline, and KV-Fresh-2-baseline with different sizes of query range. We can see that the verification cost of all mechanisms sharply increase as the number of queried keys increases. Similar to Fig. 4.12a and Fig. 4.12b, both KV-Fresh-1 and KV-Fresh-2 outperform corresponding KV-Fresh-1-baseline and KV-Fresh-2-baseline in terms of verification cost, which is expected. These results further confirm the high efficiency of KV-Fresh in processing range queries.

4.6 Summary

In this chapter, we embrace edge computing paradigm for low-latency spectrum access requests processing by outsourcing spectrum access requests processing to distributed edge servers. We have mapped the problem of authenticating outsourced spectrum access requests processing as authenticated query processing over multi-version key-value store, and presented the design and evaluation of KV-Fresh, a novel freshness authentication scheme for outsourced spectrum availability updates modeled as multi-version key-value stores. KV-Fresh is built upon LKS-MHT, a novel data structure that allows efficient proof of no update over a potentially large number of intervals. KV-Fresh supports both point query and range query. Extensive simulation studies confirm that KV-Fresh can always simultaneously achieve strong real-time guarantee and high communication efficiency.

Chapter 5

CONCLUSION AND FUTURE WORK

In this dissertation, we have tackled three key security and privacy challenges in database-driven DSS to pave the way for its wide development and deployment. First, we introduce a novel mechanism that allows a DBA to construct highly accurate REMs in the presence of false spectrum measurements. Inspired by self-labeled techniques, our solution iteratively constructs an REM using a small number of trusted measurements and gradually incorporating measurements from mobile sensors by jointly considering each measurements spatial fitness of trusted measurements and the long-term behavior of the mobile sensor. We have confirmed the effectiveness of our solution via detailed simulation studies using a real spectrum measurement dataset.

Second, we present a novel differentially-private reverse auction mechanism to stimulate mobile workers' participation in crowdsourcing-based REM construction by integrating a novel greedy algorithm for winner selection with differential privacy. Through a combination of theoretical analysis and simulation studies using real spectrum measurements, we have confirmed that the proposed incentive mechanism can simultaneously achieve differential bid privacy, approximated truthfulness, individual rationality, budget feasibility, and high REM accuracy.

Third, we explore the edge computing paradigm for low-latency spectrum-access requests processing and study the problem of authenticated spectrum-access requests processing via untrusted edge servers. By mapping the problem into authenticated outsourced multi-version key-value stores, we propose KV-Fresh, a novel freshness authentication scheme based on LKS-MHT, a novel data structure that allows efficient proof of no update over a potentially large number of intervals. Our solution supports both point query and range query. Extensive simulation studies using a real dataset

show that KV-Fresh is not only much more efficient but also offers stronger real-time guarantee than state-of-the-art freshness authentication techniques.

There are a number of issues worthy of further investigation. First, we notice that our solution for secure REM construction works in an iterative fashion which may incur high computation latency when processing a large number of spectrum measurements. We therefore plan to investigate alternative solutions with low computation complexity. Second, while our simulation results suggest that our differentially-private reverse auction mechanism is truthful, we have only been able to prove that its approximate truthfulness. We will seek to either prove its truthfulness or develop alternative solutions to guarantee truthfulness. Last but not least, we plan to extend KV-Fresh to support other types of non-SQL database such as document store.

REFERENCES

[1] FCC, "Spectrum policy task force report," https://transition.fcc.gov/sptf/files/SEWGFinalReport_1.pdf, 2002.

[2] ——, "National broadband plan," https://transition.fcc.gov/national-broadband-plan/national-broadband-plan.pdf, 2010.

[3] President's Council of Advisors on Science and Technology, "Realizing the full potential of government-held spectrum to spur economic growth," https://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast_spectrum_report_final_july_20_2012.pdf, 2012.

[4] D. Gurney, G. Buchwald, L. Ecklund, S. L. Kuffner, and J. Grosspietsch, "Geo-location database techniques for incumbent protection in the tv white space," in *DySPAN'08*, Oct 2008, pp. 1–9.

[5] R. Murty, R. Chandra, T. Moscibroda, and P. Bahl, "Senseless: A database-driven white spaces network," *IEEE Transactions on Mobile Computing*, vol. 11, no. 2, pp. 189–203, Feb 2012.

[6] "Second report and order and memorandum opinion and order," FCC, 2008.

[7] A.G.Longley and P.L.Rice, "Prediction of tropospheric radio transmission loss over irregular terrain. a computer method," OTIC Document, Tech. Rep., 1968.

[8] T. Zhang and S. Banerjee, "Inaccurate spectrum databases?: Public transit to its rescue!" in *HotNets'13*, College Park, Maryland, 2013, pp. 6:1–6:7.

[9] T. Zhang, N. Leng, and S. Banerjee, "A vehicle-based measurement framework for enhancing whitespace spectrum databases," in *MobiCom'14*, Maui, Hawaii, USA, 2014, pp. 17–28.

[10] A. Chakraborty and S. R. Das, "Measurement-augmented spectrum databases for white space spectrum," in *CoNEXT'14*, Sydney, Australia, 2014, pp. 67–74.

[11] A. Saeed, K. A. Harras, and M. Youssef, "Towards a characterization of white spaces databases errors: An empirical study," in *WiNTECH'14*, Maui, Hawaii, USA, 2014, pp. 25–32.

[12] Y. Zhao, "Enabling cognitive radios through radio environment maps," Ph.D. dissertation, Virginia Polytechnic Institute and State University, May 2007.

[13] X. Ying, S. Roy, and R. Poovendran, "Incentivizing crowdsourcing for radio environment mapping with statistical interpolation," in *IEEE DySPAN'15*, Stockholm, Sweden, Sept 2015, pp. 365–374.

[14] B. Gao, S. Bhattarai, J. J. Park, Y. Yang, M. Liu, K. Zeng, and Y. Dou, "Incentivizing spectrum sensing in database-driven dynamic spectrum sharing," in *IEEE INFOCOM'16*, April 2016, pp. 1–9.

[15] R. Chen, J. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *IEEE INFOCOM'08*, Phoenix, AZ, USA, April 2008, pp. 1876–1884.

[16] A. Min, K. Shin, and X. Hu, "Attack-tolerant distributed sensing for dynamic spectrum access networks," in *ICNP'09*, Princeton, NJ, oct. 2009, pp. 294–303.

[17] H. Li and Z. Han, "Catch me if you can: An abnormality detection approach for collaborative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3554–3565, Nov. 2010.

[18] O. Fatemieh, R. Chandra, and C. Gunter, "Secure collaborative sensing for crowdsourcing spectrum data in white space networks," in *DySPAN'10*, Singapore, Apr. 2010.

[19] O. Fatemieh, A. Farhadi, R. Chandra, and C. Gunter, "Using classification to protect the integrity of spectrum measurements in white space networks," in *NDSS'11*, San Diego, CA, Feb. 2011.

[20] K. Zeng, P. Paweczak, and D. Cabric, "Reputation-based cooperative spectrum sensing with trusted nodes assistance," *IEEE Communications Letters*, vol. 14, no. 3, pp. 226–228, march 2010.

[21] O. Fatemieh, M. LeMay, and C. Gunter, "Reliable telemetry in white spaces using remote attestation," in *ACSAC'11*, Orlando, FL, 2011, pp. 323–332.

[22] S. Choi and K. G. Shin, "Secure cooperative spectrum sensing in cognitive radio networks using interference signatures," in *IEEE CNS'13*, National Harbor, MD, USA, Oct 2013, pp. 19–27.

[23] R. Zhang, J. Zhang, Y. Zhang, and C. Zhang, "Secure crowdsourcing-based cooperative spectrum sensing," in *INFOCOM'13*, Turin, Italy, Apr. 2013.

[24] W. Wang, L. Chen, K. G. Shin, and L. Duan, "Secure cooperative spectrum sensing and access against intelligent malicious behaviors," in *IEEE INFOCOM'14*, April 2014, pp. 1267–1275.

[25] ——, "Thwarting intelligent malicious behaviors in cooperative spectrum sensing," *IEEE Transactions on Mobile Computing*, vol. 14, no. 1, pp. 2392–2405, 2015.

[26] A. Abrardo, M. Barni, K. Kallas, and B. Tondi, "A game-theoretic framework for optimum decision fusion in the presence of byzantines," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1333–1345, June 2016.

[27] A. B. H. Alaya-Feki, S. B. Jemaa, B. Sayrac, P. Houze, and E. Moulines, "Informed spectrum usage in cognitive radio networks: Interference cartography," in *PIMRC'08*, Sept 2008, pp. 1–5.

[28] A. Achtzehn, J. Riihijärvi, G. M. Vargas, M. Petrova, and P. Mahonen, "Improving coverage prediction for primary multi-transmitter networks operating in the tv whitespaces," in *IEEE SECON'12*, Seoul, South Korea, Aug. 2012, pp. 623–631.

[29] C. Phillips, M. Ton, D. Sicker, and D. Grunwald, "Practical radio environment mapping with geostatistics," in *IEEE DYSPAN'12*, Oct 2012, pp. 422–433.

[30] X. Ying, C. W. Kim, and S. Roy, "Revisiting tv coverage estimation with measurement-based statistical interpolation," in *COMSNETS'15*, Bangalore, India, Jan 2015, pp. 1–8.

[31] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study," *Knowledge and Information Systems*, vol. 42, no. 2, pp. 245–284, 2015.

[32] Y. Zhao, B. Le, and J. H. Reed, "Chapter 11 - network support: The radio environment map," in *Cognitive Radio Technology*, B. A. Fette, Ed., 2006, pp. 337 – 363.

[33] Y. Zhao, L. Morales, J. Gaeddert, K. K. Bae, J. S. Um, and J. H. Reed, "Applying radio environment maps to cognitive wireless regional area networks," in *DySPAN'07*, April 2007, pp. 115–118.

[34] H. B. Yilmaz, T. Tugcu, F. Alagoz, and S. Bayhan, "Radio environment map as enabler for practical cognitive radio networks," *IEEE Communications Magazine*, vol. 51, no. 12, pp. 162–169, 2013.

[35] A. Min, X. Zhang, and K. Shin, "Detection of small-scale primary users in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 2, pp. 349 –361, Feb. 2011.

[36] A. Nika, Z. Zhang, X. Zhou, B. Y. Zhao, and H. Zheng, "Towards commoditized real-time spectrum monitoring," in *HotWireless'14*, Maui, HI, 2014, pp. 25–30.

[37] R. Calvo-Palomino, D. Pfammatter, D. Giustiniano, and V. Lenders, "A low-cost sensor platform for large-scale wideband spectrum monitoring," in *IPSN'15*, Seattle, Washington, 2015, pp. 396–397.

[38] D. Pfammatter, D. Giustiniano, and V. Lenders, "A software-defined sensor architecture for large-scale wideband spectrum monitoring," in *IPSN'15*, Seattle, Washington, 2015, pp. 71–82.

[39] X. Liu, F. Chen, and C. T. Lu, "Robust prediction and outlier detection for spatial datasets," in *IEEE ICDM'12*, Dec 2012, pp. 469–478.

[40] M. Pesko, T. Javornik, A. Kosir, M. Stular, and M. Mohorcic, "Radio environment maps: The survey of construction methods," *KSII Transactions on Internet and Information Systems*, vol. 8, no. 11, pp. 3789–3809, 2014.

[41] A. Achtzehn, J. Riihijarvi, and P. Mahonen, "Improving accuracy for tvws geolocation databases: Results from measurement-driven estimation approaches," in *IEEE DySPAN'14*, McLean, VA, USA, April 2014, pp. 392–403.

[42] J. Ojaniemi, J. Kalliovaara, J. Poikonen, and R. Wichman, "A practical method for combining multivariate data in radio environment mapping," in *PIMRC'13*, Sept 2013, pp. 729–733.

[43] Y. Dai and J. Wu, "Integration of spectrum database and sensing results for hybrid spectrum access systems," in *MASS'15*, Oct 2015, pp. 28–36.

[44] W. Wang, H. Li, Y. Sun, and Z. Han, "Catchit: Detect malicious nodes in collaborative spectrum sensing," in *IEEE GLOBECOM'09*, Honolulu, HI, USA, March 2009.

[45] H. Chen, M. Zhou, L. Xie, and J. Li, "Cooperative spectrum sensing with m-ary quantized data in cognitive radio networks under ssdf attacks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 8, pp. 5244–5257, 2017.

[46] A. S. Rawat, P. Anand, H. Chen, and P. K. Varshney, "Collaborative spectrum sensing in the presence of byzantine attacks in cognitive radio networks," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 774–786, 2011.

[47] H. Chen, M. Zhou, L. Xie, K. Wang, and J. Li, "Joint spectrum sensing and resource allocation scheme in cognitive radio networks with spectrum sensing data falsification attack," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 11, pp. 9181–9191, 2016.

[48] S. Yang, F. Wu, S. Tang, X. Gao, B. Yang, and G. Chen, "On designing data quality-aware truth estimation and surplus sharing method for mobile crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 832–847, 2017.

[49] J. Hu, H. Lin, X. Guo, and J. Yang, "Dtcs: An integrated strategy for enhancing data trustworthiness in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4663–4671, Dec 2018.

[50] C. Miao, Q. Li, H. Xiao, W. Jiang, M. Huai, and L. Su, "Towards data poisoning attacks in crowd sensing systems," in *ACM MobiHoc'18*, Los Angeles, CA, USA, June 2018, pp. 111–120.

[51] C. Miao, Q. Li, L. Su, M. Huai, W. Jiang, and J. Gao, "Attack under disguise: An intelligent data poisoning attack mechanism in crowdsourcing," in *WWW'18*, Lyon, France, April 2018, pp. 13–22.

[52] T. Dierks and E. Rescorla, "The transport layer security (TLS) protocol," RFC 4346, Apr. 2006.

[53] Y. Liu, P. Ning, and H. Dai, "Authenticating primary users' signals in cognitive radio networks via integrated cryptographic and wireless link signatures," in *S&P'10*, Washington, DC, USA, 2010, pp. 286–301.

[54] N. Cressie, *Statistics for Spatial Data*. John Wiley & Sons, 1993.

[55] A. Konak, "A kriging approach to predicting coverage in wireless networks," *Int. J. Mob. Netw. Des. Innov.*, vol. 3, no. 2, pp. 65–71, Jan. 2009.

[56] H. Braham, S. B. Jemaa, B. Sayrac, G. Fort, and E. Moulines, "Low complexity spatial interpolation for cellular coverage analysis," in *WiOpt'14*, May 2014, pp. 188–195.

[57] R. A. Olea, "A six-step practical approach to semivariogram modeling," *Stochastic Environmental Research and Risk Assessment*, vol. 20, no. 5, pp. 307–318, Jul 2006.

[58] N. Cressie, "Fitting variogram models by weighted least squares," *Mathematical Geology*, vol. 17, no. 5, pp. 565–586, 1985.

[59] M. Ton and C. Phillips, "CRAWDAD dataset cu/wimax (v. 2012-06-01)," Downloaded from http://crawdad.org/cu/wimax/20120601, Jun. 2012.

[60] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *FOCS'07*, Washington, DC, USA, 2007, pp. 94–103.

[61] D. Yang, G. Xue, X. Fang, and J. Tang, "Crowdsourcing to smartphones: Incentive mechanism design for mobile phone sensing," in *MobiCom'12*, Istanbul, Turkey, Aug. 2012.

[62] D. Zhao, X.-Y. Li, and H. Ma, "How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint," in *IEEE INFOCOM'14*, Toronto, Canada, Apr. 2014.

[63] X. Jin and Y. Zhang, "Privacy-preserving crowdsourced spectrum sensing," in *IEEE INFOCOM'16*, San Francisco, CA, USA, July 2016.

[64] T. Wen, Y. Zhu, and T. Liu, "P2: A location privacy-preserving auction mechanism for mobile crowd sensing," in *IEEE GLOBECOM'16*, Washington, DC, USA, Dec. 2016.

[65] J. Lin, D. Yang, , M. Li, J. Xu, and G. Xue, "Frameworks for privacy-preserving mobile crowdsensing incentive mechanisms," *IEEE Transactions on Mobile Computing*, 2017.

[66] K. Nissim, R. Smorodinsky, and M. Tennenholtz, "Approximately optimal mechanism design via differential privacy," in *ITCS'12*, Cambridge, MA, 2012, pp. 203–213.

[67] Y. Chen, S. Chong, I. A. Kash, T. Moran, and S. Vadhan, "Truthful mechanisms for agents that value privacy," in *EC'13*, Philadelphia, PA, 2013, pp. 215–232.

[68] H. Huang, X. Y. Li, Y. e. Sun, H. Xu, and L. Huang, "PPS: Privacy-preserving strategyproof social-efficient spectrum auction mechanisms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1393–1404, May 2015.

[69] Q. Huang, Y. Tao, and F. Wu, "SPRING: a strategy-proof and privacy preserving spectrum auction mechanism," in *INFOCOM'13*, April 2013, pp. 827–835.

[70] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov¡ì, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *ICDCS'16*, Nara, Japan, June 2016.

[71] S. Leung and E. Lui, "Bayesian mechanism design with efficiency, privacy, and approximate truthfulness," in *WINE'12*, Berlin, Heidelberg, 2012, pp. 58–71.

[72] D. Xiao, "Is privacy compatible with truthfulness?" in *ITCS'13*, Berkeley, CA, 2013, pp. 67–86.

[73] M. Pan, J. Sun, and Y. Fang, "Purging the back-room dealing: Secure spectrum auction leveraging paillier cryptosystem," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 4, pp. 866–876, April 2011.

[74] R. Zhu, Z. Li, F. Wu, K. Shin, and G. Chen, "Differentially private spectrum auction with approximate revenue maximization," in *MobiHoc'14*, Aug. 2014, pp. 185–194.

[75] R. Zhu and K. G. Shin, "Differentially private and strategy-proof spectrum auction with approximate revenue maximization," in *INFOCOM'15*, Apr. 2015, pp. 918–926.

[76] H. Jin, L. Su, B. Ding, K. Nahrstedt, and N. Borisov, "Enabling privacy-preserving incentives for mobile crowd sensing systems," in *IEEE ICDCS'16*, June 2016, pp. 344–353.

[77] Z. Feng, Y. Zhu, Q. Zhang, L. M. Ni, and A. V. Vasilakos, "Trac: truthful auction for location-aware collaborative sensing in mobile crowdsoucing," in *INFOCOM'14*, Apr. 2014, pp. 1231–1239.

[78] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "INCEPTION: incentivizing privacy-preserving data aggregation for mobile crowd sensing systems," in *ACM MobiHoc '16*, Paderborn, Germany, July 2016, pp. 341–350.

[79] H. Jin, L. Su, and K. Nahrstedt, "CENTURION: incentivizing multi-requester mobile crowd sensing," in *IEEE INFOCOM'17*, Atlanta, GA, May 2017, pp. 1–9.

[80] Y. Hu and R. Zhang, "Secure crowdsourced radio environment map construction against false spectrum measurements," in *IEEE ICNP'17*, Toronto, Canada, Oct. 2017.

[81] C. Dwork, "Differential privacy," in *ICALP'06*, Venice, Italy, July 2006, pp. 1–12.

[82] S. Rathinakumar and M. K. Marina, "Gavel: Strategy-proof ascending bid auction for dynamic licensed shared access," in *MobiHoc'16*, Paderborn, Germany, July 2016, pp. 121–130.

[83] A. Das and D. Kempe, "Algorithms for subset selection in linear regression," in *STOC'08*, Victoria, British Columbia, Canada, 2008, pp. 45–54.

[84] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 03 1951.

[85] H. Hacigumus, B. Iyer, and S. Mehrotra, "Providing database as a service," in *ICDE'02*, San Jose, CA, USA, Feb 2002.

[86] M. Narasimha and G. Tsudik, "Authentication of outsourced databases using signature aggregation and chaining," in *DASFAA'06*, Singapore, Apr. 2006, pp. 420–436.

[87] H. Pang and K.-L. Tan, "Verifying completeness of relational query answers from online servers," *ACM Trans. Inf. Syst. Secur.*, vol. 11, no. 2, pp. 1–50, 2008.

[88] H. Pang, J. Zhang, and K. Mouratidis, "Scalable verification for outsourced dynamic databases," *Proc. VLDB Endow.*, vol. 2, no. 1, pp. 802–813, 2009.

[89] Y. Yang, S. Papadopoulos, D. Papadias, and G. Kollios, "Authenticated indexing for outsourced spatial databases," *The VLDB Journal*, vol. 18, no. 3, pp. 631–648, Jun. 2009.

[90] H. Hu, J. Xu, Q. Chen, and Z. Yang, "Authenticating location-based services without compromising location privacy," in *SIGMOD'12*, Scottsdale, AZ, May 2012, pp. 301–312.

[91] X. Lin, J. Xu, and H. Hu, "Authentication of location-based skyline queries," in *CIKM'11*. New York, NY: ACM, Oct. 2011, pp. 1583–1588.

[92] X. Lin, J. Xu, and J. Gu, "Continuous skyline queries with integrity assurance in outsourced spatial databases," in *WAIM'12*. Harbin, China: Springer, Aug. 2012, pp. 114–126.

[93] X. Lin, J. Xu, H. Hu, and W.-C. Lee, "Authenticating location-based skyline queries in arbitrary subspaces," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 6, pp. 1479–1493, June 2014.

[94] Q. Chen, H. Hu, and J. Xu, "Authenticating top-k queries in location-based services with confidentiality," *Proceedings of the VLDB Endowment*, vol. 7, no. 1, pp. 49–60, Sep. 2013.

[95] R. Zhang, Y. Zhang, and C. Zhang, "Secure top-k query processing via untrusted location-based service providers," in *INFOCOM'12*, Orlando, FL, Mar. 2012.

[96] R. Zhang, J. Sun, Y. Zhang, and C. Zhang, "Secure spatial top-k query processing via untrusted location-based service providers," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 111–124, Jan 2015.

[97] F. Li, M. Hadjieleftheriou, G. Kollios, and L. Reyzin, "Dynamic authenticated index structures for outsourced databases," in *SIGMOD '06*, Chicago, IL, 2006, pp. 121–132.

[98] F. Li, K. Yi, M. Hadjieleftheriou, and G. Kollios, "Proof-infused streams: Enabling authentication of sliding window queries on streams," in *VLDB'07*, Vienna, Austria, Sep. 2007, pp. 147–158.

[99] E. Stefanov, M. van Dijk, A. Juels, and A. Oprea, "Iris: A scalable cloud file system with efficient integrity checks," ser. ACSAC '12, December 2012, pp. 229–238.

[100] H.-J. Yang, V. Costan, N. Zeldovich, and S. Devadas, "Authenticated storage using small trusted hardware," ser. CCSW '13, 2013, pp. 35–46.

[101] Y. Tang, T. Wang, L. Liu, X. Hu, and J. Jang, "Lightweight authentication of freshness in outsourced key-value stores," in *Proceedings of the 30th Annual Computer Security Applications Conference*, ser. ACSAC '14, New Orleans, Louisiana, USA, 2014, pp. 176–185.

[102] S. Papadopoulos, Y. Yang, and D. Papadias, "Cads: Continuous authentication on data streams," ser. VLDB '07, Sep. 2007, pp. 135–146.

[103] M. T. Goodrich, C. Papamanthou, R. Tamassia, and N. Triandopoulos, "Athos: Efficient authentication of outsourced file systems," in *Information Security*, 2008, pp. 80–96.

[104] A. J. Feldman, W. P. Zeller, M. J. Freedman, and E. W. Felten, "Sporc: Group collaboration using untrusted cloud resources," ser. OSDI'10, Oct. 2010, pp. 337–350.

[105] R. A. Popa, C. M. S. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: Protecting confidentiality with encrypted query processing," ser. SOSP '11, Oct. 2011, pp. 85–100.

[106] P. Mahajan, S. Setty, S. Lee, A. Clement, L. Alvisi, M. Dahlin, and M. Walfish, "Depot: Cloud storage with minimal trust," *ACM Trans. Comput. Syst.*, vol. 29, no. 4, pp. 12:1–12:38, Dec. 2011.

[107] M. L. Yiu, E. Lo, and D. Yung, "Authentication of moving knn queries," in *ICDE'11*. Hannover, Germany: IEEE, Apr. 2011, pp. 565–576.

[108] L. Hu, W.-S. Ku, S. Bakiras, and C. Shahabi, "Spatial query integrity with voronoi neighbors," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 4, pp. 863–876, Apr. 2013.

[109] M. L. Yiu, Y. Lin, and K. Mouratidis, "Efficient verification of shortest path search via authenticated hints," in *ICDE'10*. Long Beach, CA: IEEE, Mar. 2010, pp. 237–248.

[110] L. Harn, "Batch verifying multiple rsa digital signatures," *Electronics Letters*, vol. 34, no. 12, pp. 1219–1220, June 1998.

[111] TrueFax, "January 2019 historical tick-by-tick data," Downloaded from https://www.truefx.com/?page=download&description=january2019&dir=2019/2019-01, Jan 2019.

Appendix A

PERMISSIONS

Internal or personal use of IEEE/ACM copyrighted materials involved in this dissertation is permitted.

Chapter 2 is based on the two papers:

Chapter 3 is based on one paper:

Chapter 4 is based on one paper: